罗克韦尔
PLC控制技术



主　编　王德吉
副主编　申玉军　黄光富





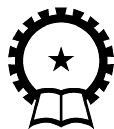
机械工业出版社
CHINA MACHINE PRESS

VISIT…

LANZAROTE
Caliente.COM

罗克韦尔 PLC 控制技术

主　编　王德吉
副主编　申玉军　黄光富



机 械 工 业 出 版 社

本书围绕罗克韦尔自动化公司控制平台的核心技术从不同角度予以阐述。通过编程软件的操作过程，展示了大量组态界面。在阐述参数设置定义和含义的同时，说明了产品的功能和作用；通过产品平台网络及通信方式的讨论，描述了数据的交换过程；通过高级运用的介绍，拓展了标准化编程的理念；通过第三方和传统产品的运用实例，显现了产品的高度兼容。

本书主要内容有：Logix 入门、SFC 编程、功能块编程、用户自定义指令 AOI、ME PowerFlex 面板、变频器的首要集成和集成架构构建器。结合产品特点，循序渐进、由浅入深地介绍了产品的运用，是一本概念清晰、实用性很强的实训教程。

本书可以作为自动化项目开发工程技术人员和现场维护人员手边的宝典，还可作为大专院校相关专业学生走向实践工作的指导用书。

图书在版编目（CIP）数据

罗克韦尔 PLC 控制技术/王德吉主编. —北京：机械工业出版社，2014.8
ISBN 978 -7 -111 -47798 -3

Ⅰ.①罗… Ⅱ.①王… Ⅲ.①plc 技术 - 教材 Ⅳ.①TM571.6

中国版本图书馆 CIP 数据核字（2014）第 199146 号

机械工业出版社（北京市百万庄大街2 2 号 邮政编码 100037
策划编辑：林春泉 责任编辑：张沪光
版式设计：霍永明 责任校对：任秀丽 胡艳萍
责任印制：常天培
北京京丰印刷厂印刷
2 0 1年51 月第 1 版 · 第 1 次印刷
184mm×260mm · 19印张 · 510千字
3001—4000册
标准书号：ISBN 978 -7 -111 -47798 -3
定价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社服务中心：(010)88361066	教材网：http://www.cmpedu.com
销售一部：(010)68326294	机工官网：http://www.cmpbook.com
销售二部：(010)88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：(010)88379203	**封面无防伪标均为盗版**

序

随着微处理器技术、电力电子技术、网络技术、控制技术的发展，PLC 系统实现了全数字化、智能化、网络化，已成为自动控制系统发展的主流方向。PLC 控制技术在先进制造领域以及在提高生产速度、管理生产过程、合理高效加工和保证安全生产等方面起到越来越关键的作用。

烟草行业是全球各生产制造行业中自动化程度比较高的行业之一。世界各地的许多卷烟厂都热衷于自动化技术改造，无论是欧美发达国家，还是亚洲新兴工业化国家的卷烟生产企业，在自动化技术改造方面都不遗余力。中国烟草企业从 20 世纪 80 年代初期引进国外先进的生产设备开始，就在不断探索和追踪最新的自动化技术。目前，烟草企业对罗克韦尔 PLC 技术学习的呼声越来越强烈。

我院作为中国烟草行业技术人才的专门培训机构，多年来一直致力于追踪全球工业自动化领域最新的技术和理念并将其转化为生产力，服务于国内的烟草企业，并为之培养、培训了大批的技术及技术管理人才。近年来，为了满足烟草企业针对工业控制方面的特殊培训需求，我院与罗克韦尔公司展开了多层次，全方位的人才培训合作。《罗克韦尔 PLC 控制技术》一书就是罗克韦尔公司与我院机电工程研究室合作的结晶。该书将罗克韦尔 PLC 技术汇集其中，并结合现场应用实例，将罗克韦尔 PLC 技术详尽地呈现给读者。希望本书能为自动化领域设计人员和企业工程师及院校师生提供有力的支持和帮助，也请广大读者为该书提出批评和修正意见。

中国烟草总公司职工进修学院

李广才　副院长

2014 年 7 月

前　言

随着计算机技术的迅猛发展，工业控制产品加快了推陈出新的步伐，可编程序控制器问世以来，已逐步发展成为操作方便、功能齐全、组合自由、网络开放、数据标准、兼容性强、覆盖面广的新一代控制产品，罗克韦尔自动化公司推出的ControlLogix控制平台就是极具代表性的新型产品。《罗克韦尔PLC控制技术》围绕ControlLogix控制平台的核心技术从不同角度予以讨论。通过编程软件的操作过程，展示了大量组态界面，在阐述参数设置定义和含义的同时，说明了产品的功能和作用；通过产品平台网络及通信方式的讨论，描述了数据的交换过程；通过高级运用的介绍，拓展了标准化编程的理念；通过第三方和传统产品的运用实例，显现了产品的高度兼容。

《罗克韦尔PLC实训教程》一书结合产品特点，循序渐进、由浅入深地介绍了产品的运用，是一本概念清晰、实用性很强的手册。本书不但可以作为项目开发和现场维护人员手边的宝典，也可作为在校学生走向实践工作的指导书。

本书内容包括：Logix入门；Logix编程语言——FBD、ST、SFC；用户自定义指令AOI；与变频的首要集成；ME Powerflex面板；集成架构构建器IAB。

本书主编长期从事自动化与信息化研究与培训工作。先后参与国家973、国家863、自然基金等项目10余项，兼任IFAC委员、国际工程师协会委员，中科院兼职博士生导师、高级访问学者，中国科协企业创新专家，中国科技部创新研究会首批会员。先后在美国微阳研究院、微软研究院从事客座研究。先后访问德国费斯托、西门子、虹霓等公司和英国埃塞克斯大学、牛津大学。

参加编写的人员还有齐海英、陈新爽、张先哲、钱国华、王玉、栗卫军、杨彬、李源源、张旭、谢俊明、马晓、汪翠兰、王德玉、丁文萍、周雪军、张春杰、石怀忠、谢俊红、姚志超、王志国、张胜利。

在本书编写过程中，得到了罗克韦尔自动化公司相关技术人员的指导和帮助，在此表示感谢。限于作者的学识水平、时间和精力，书中可能存在疏忽和谬误之处，恳请广大读者及时指正，作者深表感谢。

罗克韦尔自动化（中国）有限公司

陈新爽

2014年7月

目　　录

序
前言
第一部分　Logix 入门 …… 1
实验 1：创建新项目 …… 1
实验 2：组态 I/O …… 12
实验 3：将计算机连接到控制器 …… 19
实验 4：将计算机中的项目加载到控制器 …… 22
实验 5：使用 RSLogix 5000 帮助演示 …… 23
实验 6：标签和数据范围划定 …… 27
实验 7：测试逻辑程序 …… 33
实验 8：在线添加逻辑和标签 …… 36
实验 9：规划 Logix 控制器中的标签数据库 …… 39
实验 10：INT 与 DINT 数据类型 …… 60
实验 11：使用任务和任务中的程序进行分段 …… 64
实验 12：理解周期性任务对连续任务的影响 …… 71
实验 13：使用部分导入/导出工具 …… 78
第二部分　SFC 编程 …… 89
实验 14：创建 SFC 程序 …… 89
实验 15：SFC 执行选项 …… 96
第三部分　功能块编程 …… 102
实验 16：功能块编程的简介 …… 102
实验 17：通过 Active X 面板练习使用 PIDE …… 118
第四部分　用户自定义指令 AOI …… 122
实验 18：向 PowerFlex 设备中写入参数 …… 122
实验 19：AOI 用来控制 PowerFlex 设备 …… 131
实验 20：在代码中使用用户自定义指令 …… 134
实验 21：管理和更新用户自定义指令 …… 147
实验 22：为 AOI 定义源代码保护 …… 152
实验 23：重用模块化对象 …… 159
第五部分　ME PowerFlex 面板 …… 160
实验 24：ME PowerFlex 的使用 …… 160
第六部分　变频器的首要集成 …… 179
实验 25：PowerFlex 70 变频器源代码级集成 …… 179
第七部分　集成架构构建器 …… 206
实验 26：IAB 基础知识动手实验 …… 206

第一部分　Logix 入门

实验 1：创建新项目

一、实验目的

在本实验中，将介绍 Logix 产品系列。

1） 创建新项目；

2） 编写梯形图逻辑；

3） 使用符号化标签名称；

4） 使用标签监视器/编辑器。

二、启动 RSLogix 5000 编程软件

在这部分实验中，将启动 RSLogix 5000 软件，通过该软件可以对处理器进行编程。双击桌面上的 RSLogix 5000 图标，启动 RSLogix 5000 软件。显示 RSLogix 5000 界面，如图 1-1 所示。



图 1-1　RSLogix 5000 界面

三、创建新控制器项目

在这部分实验中，将使用 ControlLogix、CompactLogix 1769-L35E 或 CompactLogix 1768-L43 处理器创建离线程序。

1. 在“文件”（File）菜单中，选择“新建”（New）。将显示“新建控制器”（New Controller）对话框，如图 1-2 所示。

2. 根据工作站的实验设备验证输入与下方显示相符，然后单击“确定”（OK）。ControlLogix 站配置如图 1-3 所示。

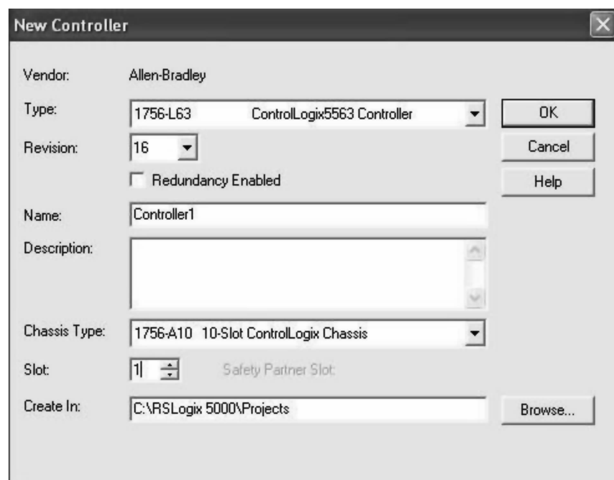


图 1-2 新建控制器（New Controller）对话框



图 1-3 ControlLogix 站配置

“新建控制器”（New Controller）

可在“新建控制器”（New Controller）窗口中定义项目。

1）“类型”（Type）：表明将使用的 Logix 控制器的类型。可以是 ControlLogix、CompactLogix、DriveLogix 或 SoftLogix 控制器。所有 Logix 控制器都仅需要一个编程软件包。

2）“版本”（Revision）：可在这里选择项目创建所使用的固件版本。当前有效的版本为 10、11、12、13、15 和 16。

3）“名称”（Name）：控制器和项目的名称。

4）“机架类型”（Chassis Type）：选择将使用的机架大小。这并不是所有类型的控制器都适用的。

5）“插槽”（Slot）：要安装控制器的插槽编号。这并不是对所有类型的控制器都适用的，例如 CompactLogix 便固定使用插槽零。

RSLogix 5000 窗口的左侧显示控制器项目管理器，其中包括一个名为 Controller Controller1 的文件夹。现在已创建首个控制器项目。此时，尚没有任何 I/O、任何标签数据库和任何与控制器关联的逻辑，如图 1-4 所示。

参考信息

控制器项目管理器是以图形方式表示控制器文件的目录。此显示界面由文件夹和文件的树组成，其中包含有关当前控制器文件中程序和数据的所有信息。此树中的默认主文件夹包括：

1）Controller File Name；

2）Tasks；

3）Motion Groups；

4）Trends；

5）Data Types；

6）I/O Configuration。

在每个文件夹前都有一个正方形，其中包含 + 或 - 符号。+ 符号表示文件夹已收起。单击该符号后将展开树并显示该文件夹中包含的文件。- 符号表示文件夹已展开，其中的内容是可见的。

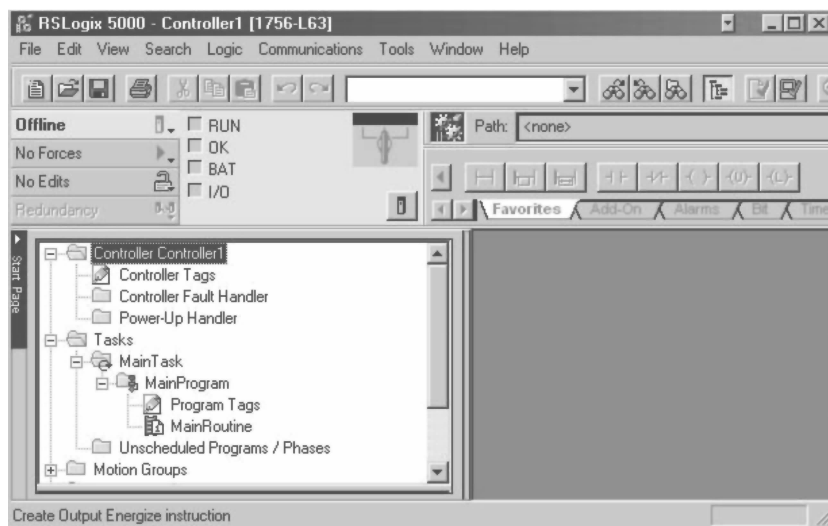


图 1-4　新的控制器项目

四、向主例程中添加梯形图逻辑

在本实验中，将添加一个简单的电动机起动/停止制动电路的代码。这将显示出使用 RSLogix 5000 软件来编程的便捷性。

在实验过程中，将仅使用梯形图逻辑来编程，但 Logix 控制器还可使用功能块、顺序功能图和结构化文本来编程。这样便可为应用选择最适合的编程语言。

将继续使用该已打开的项目。

1. 在控制器项目管理器中单击 MainProgram 文件夹的 +，以将其展开，如图 1-5 所示。

2. 展开后，将出现 MainProgram，如图 1-6 所示。

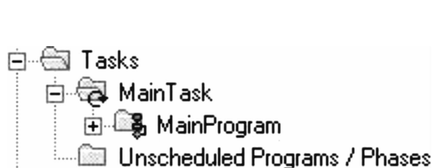


图 1-5　控制器项目管理项目列表



图 1-6　展开后的 MainProgram 项目列表

3. 双击 MainRoutine 图标（MainRoutine）。将打开例程编辑器。软件中会自动添加一个空梯级，如图 1-7 所示。

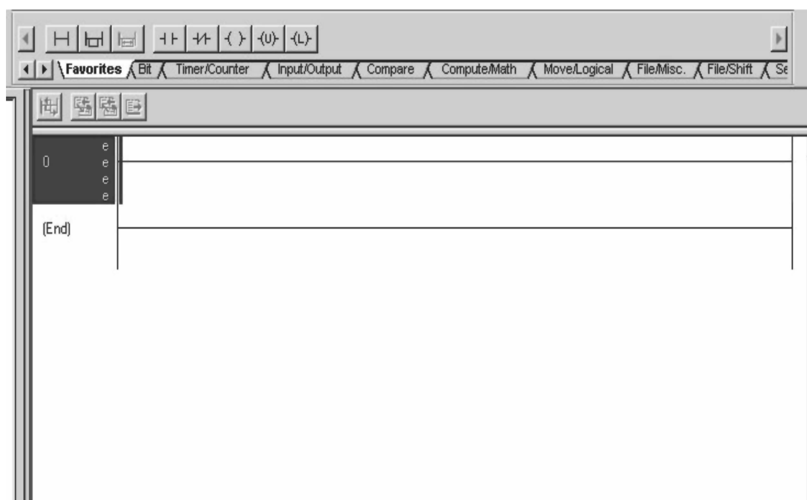


图 1-7　程序编辑器界面

4. 在指令工具栏中，左键单击并按住检查是否闭合指令 如图 1-8 所示。

5. 将 XIC 拖到梯级 0 上，直到出现如图 1-8 所示的绿色点。在想要放置指令的位置上释放鼠标按键。

6. 验证出现的梯级类似图 1-9 所示。

7. 在指令工具栏中，左键单击并按住检查是否断开指令 ，如图 1-10 所示。

8. 将 XIO 拖到梯级 0 上 XIC 指令的右侧，如图 1-10 所示。XIC 指令的右侧会再次出现绿色点，指示将插入新指令的位置。在想要放置指令的位置上释放鼠标按键。

9. 验证出现的梯级如图 1-11 所示。

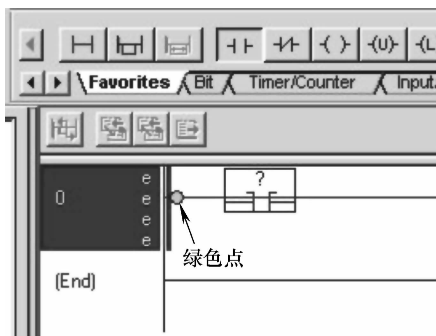


图 1-8　选择并插入闭合指令

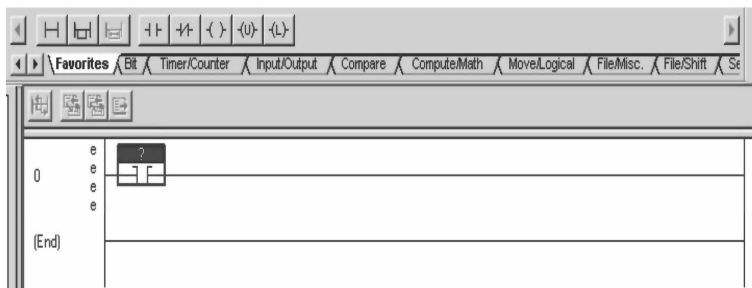


图 1-9　插入后的闭合指令



图 1-10　选择并插入断开指令

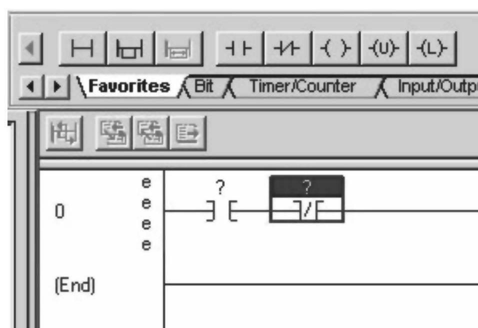


图 1-11　插入后的断开指令

参考信息

如果将指令放置在梯级上的错误位置，仅需在指令上单击并按住鼠标左键，并将该指令拖到正确的位置上即可。

10. 在指令工具栏中，左键单击并按住输出{ }激励指令，如图 1-12 所示。

11. 将 OTE 拖到梯级 0 上 XIO 指令的右侧，如图 1-12 所示。XIO 指令的右侧会再次出现绿色点，指示将插入 OTE 指令的位置。在想要插入指令的位置上释放鼠标按键，如图 1-13 所示。

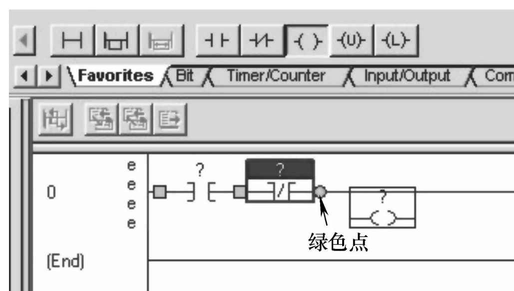


图 1-12　选择并插入输出指令

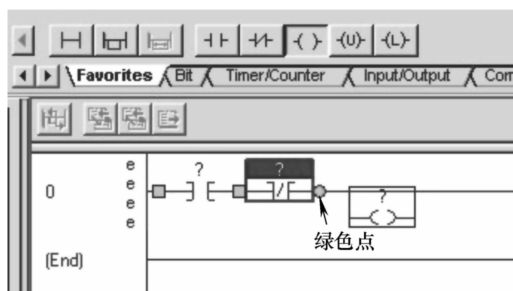


图 1-13　插入后的输出指令

12. 验证出现的梯级与图 1-14 所示相同。

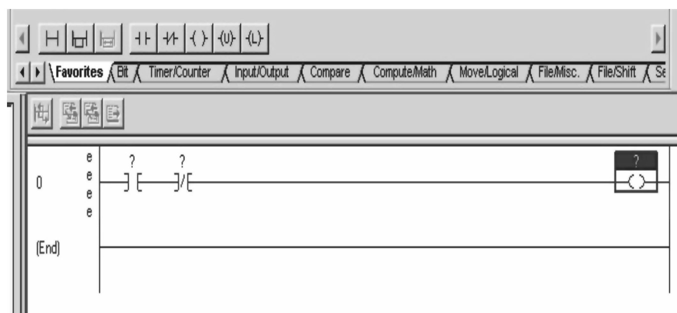


图 1-14　完成后的程序梯形图

现在，将在 XIC 指令周围添加一个分支。

13. 单击 XIC 指令，将其选中，如图 1-15 所示。

14. 在指令工具栏中单击分支指令梯级上将插入一个分支，如图 1-16 所示。

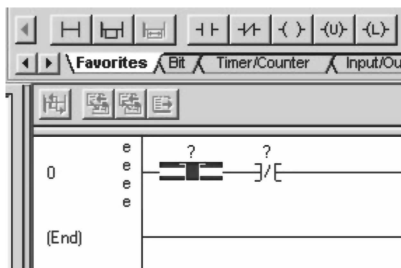


图 1-15　选中将要插入分支的指令

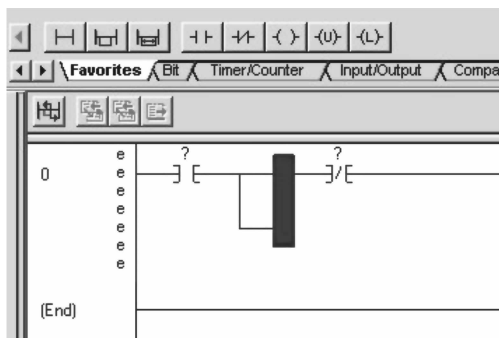


图 1-16　插入的分支标示

15. 左键单击并按住分支的蓝色突出显示部分，将选中的分支引脚拖到 XIC 指令的左侧。
16. 将分支放置到绿色点之上并释放鼠标按键，如图 1-17 所示。
17. 在指令工具栏中，左键单击并按住 XIC 指令。
18. 将 XIC 拖到新创建的分支上，直到绿色点出现。现在显示的梯级应与图 1-18 所示相同。

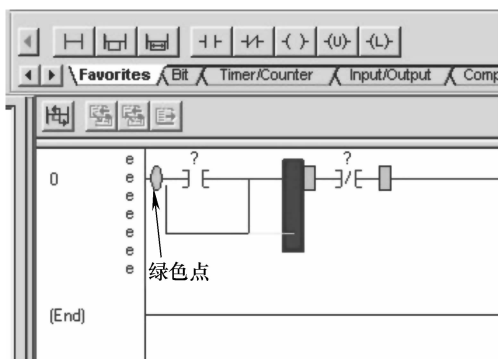


图 1-17　拖动分支标示至目标位置

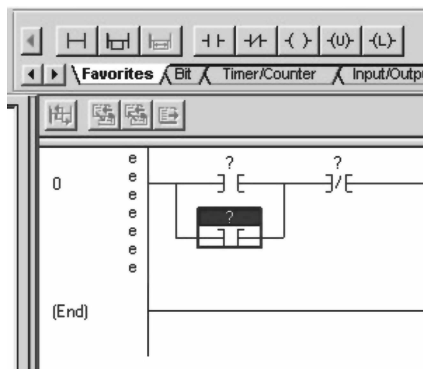


图 1-18　在分支上插入闭合指令

19. 已经完成梯级的添加，验证整个梯级类似图 1-19 所示。

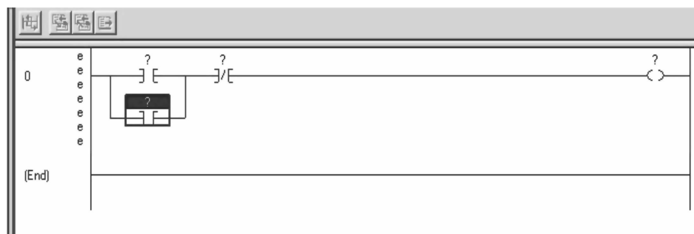


图 1-19　完成后的分支程序图

20. 在工具栏上单击“保存”（Save）图标保存程序。将程序保存到默认程序目录 C：\RSLogix 5000\Projects\中。

五、为梯形图代码创建标签

在这部分实验中，将为程序创建所需标签。在传统的 PLC 中，各数据项都用物理内存地址

标识，例如 N7:0 便是如此。而在 Logix 控制器中则没有固定的数字格式。使用标签。

将继续使用已打开的项目。

参考信息

标签是什么，为什么标签更好?

标签是内存区域的基于文本的名称。通过使用基于文本的系统，可以使用标签名称来记录梯形图代码和组织反映机器情况的数据。例如，可创建一个名为 North_Tank_Pressure 的标签，有助于加速代码生成和调试。所有标签名称都存储在控制器中。

将为该程序创建 3 个标签：即 Motor_Start、Motor_Stop 和 Motor_Run，如图 1-6 所示。

1. 首先将创建标签 Motor_Start。右键单击首个 XIC 指令的? 并选择“新建标签”（New Tag），如图 1-20 所示。

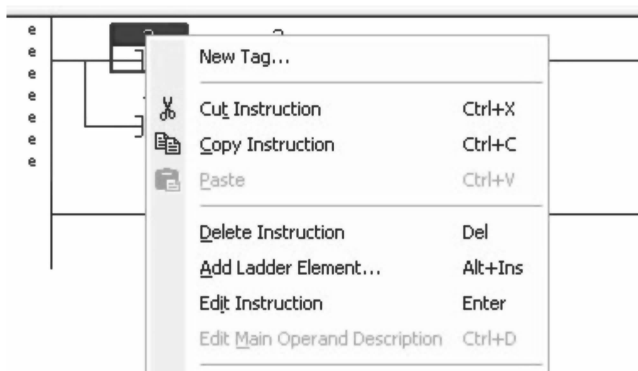


图 1-20　弹出的新建标签菜单

2. 将出现“新建标签”（New Tag）窗口，如图 1-21 所示。



图 1-21　新建标签对话窗口

参考信息

创建标签

创建标签时，共有多个标签属性。在本实验中将关注如下几个主要属性：

1）“类型”（Type）：定义标签在项目中的工作方式。

2）“基本”（Base）：存储供项目中逻辑使用的一个或多个值。

3）“别名”（Alias）：表示另一个标签的标签。

4）“生产者”（Produced）：向另一个控制器发送数据。

5）“消费者”（Consumed）：从另一个控制器接收数据。

6）“数据类型”（Data Type）：定义标签所存储数据的类型。例如，布尔型、整型、实数型、字符串型等。

7）“范围”（Scope）：定义项目中数据的访问方式。或者是以控制器为范围，即可在整个控制器内访问的全局数据；或者是以程序为范围，即特定程序的可访问数据。

3. 输入参数如图 1-22 所示。

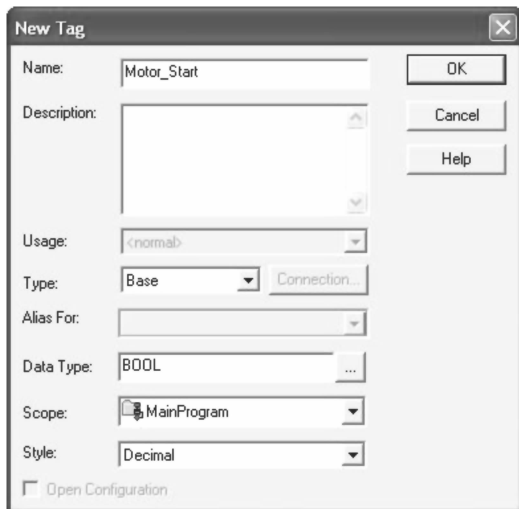


图 1-22 输入标签信息

4. 单击“确定”（OK）接受并创建标签。现在梯级将类似图 1-23。

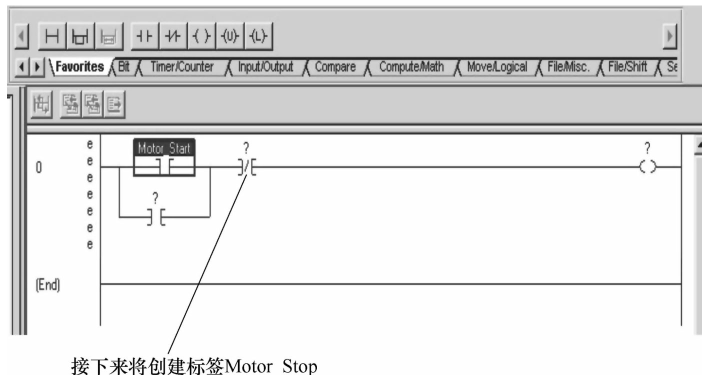


图 1-23 创建断开指令标签

5. 右键单击 XIO 指令的?，并选择“新建标签”（New Tag）。将再次出现“新建标签”（New Tag）窗口，图 1-24 所示。

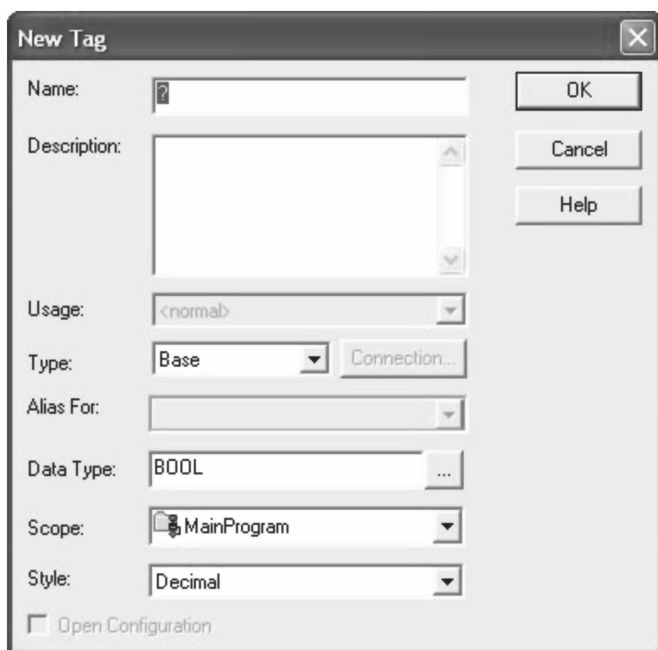


图 1-24 创建标签对话窗口

6. 输入参数，如图 1-25 所示。

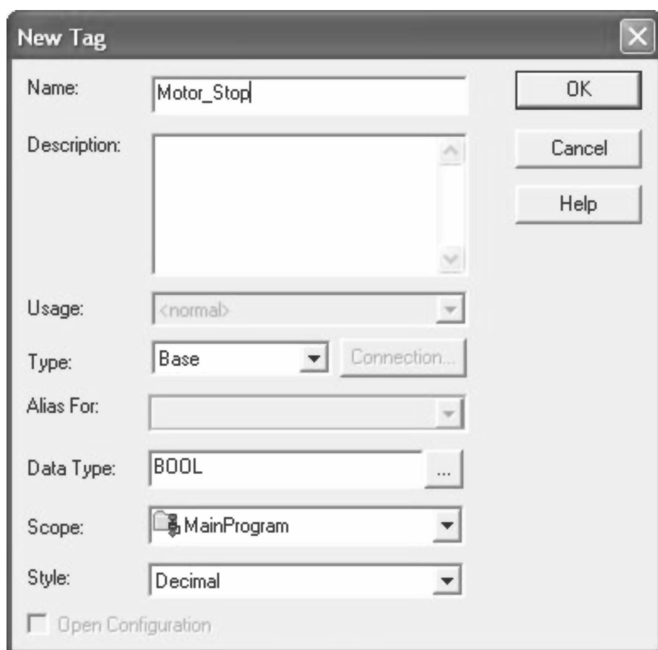


图 1-25 输入“断开”指令标签信息

7. 单击“确定”（OK）接受并创建标签。

8. 验证出现的梯级类似图 1-26 所示。

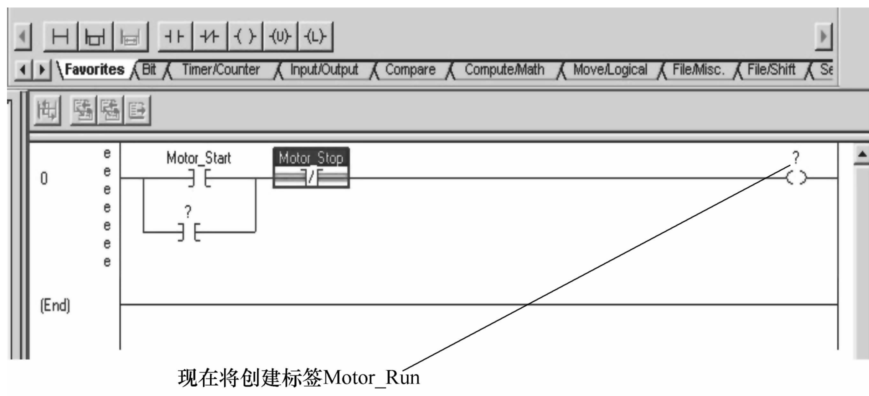


图 1-26　创建输出指令标签

9. 右键单击 OTE 指令的?，并选择“新建标签”（New Tag）。将出现“新建标签”（New Tag）窗口。

10. 输入参数如图 1-27 所示。

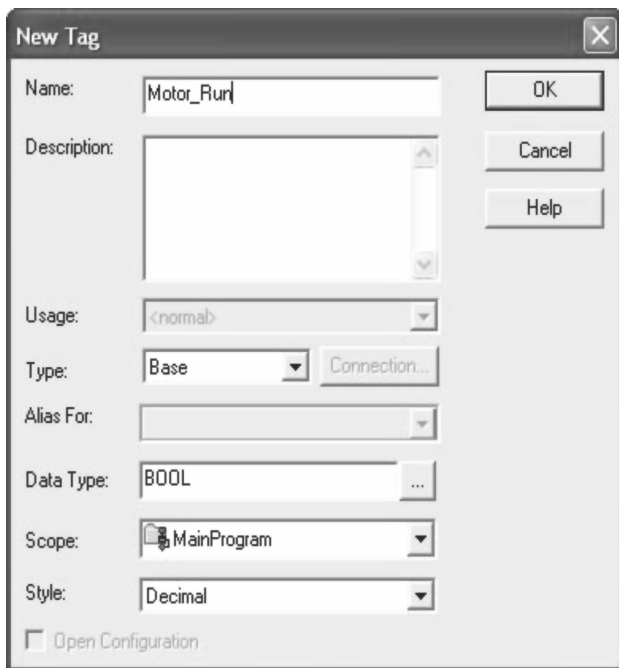


图 1-27　输入“输出”指令标签信息

11. 单击“确定”（OK）接受并创建标签。现在显示的梯级应与图 1-28 所示相同。

12. 在 OTE 指令的标签 Motor_Run 上单击并按住鼠标左键，如图 1-29 所示。

13. 将标签 Motor_Run 拖到 XIC 指令上，直到? 旁出现绿色点（见图 1-30）。然后，释放鼠标按键。现在显示的梯级应与图 1-30 相同。

请注意梯级零旁边的“e”。这些“e”表示该梯级处于编辑模式。

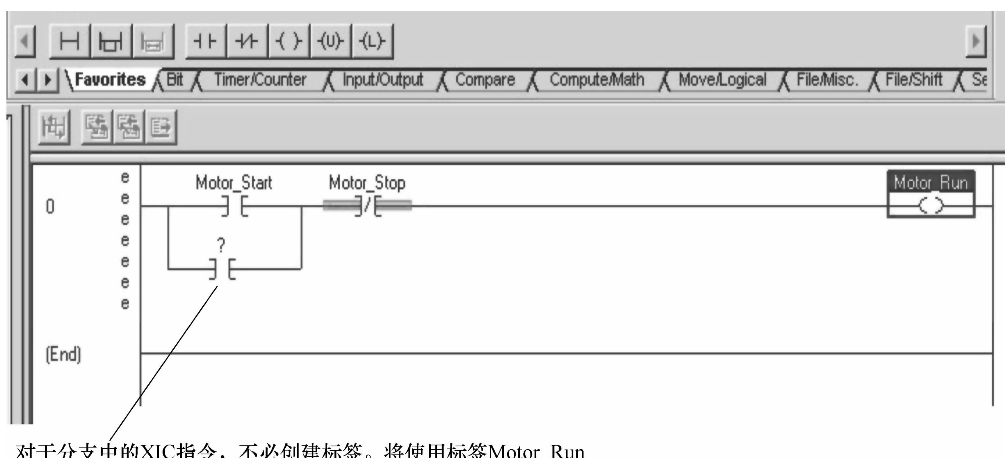


图 1-28 自锁指令标签的使用

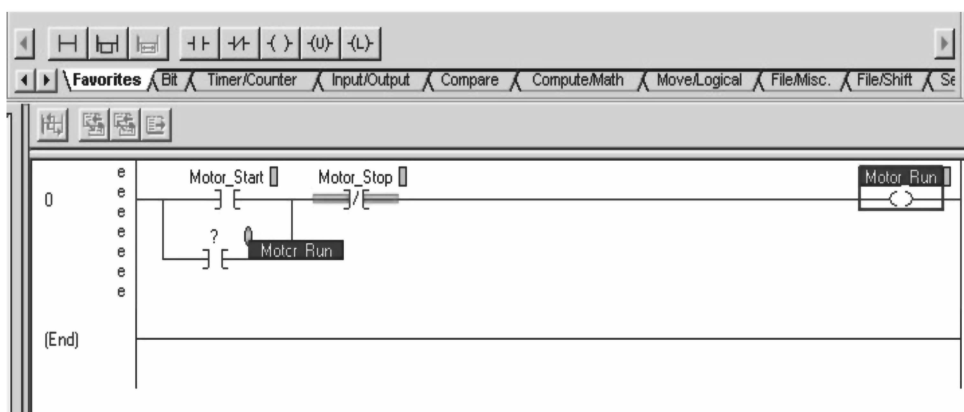


图 1-29 选中将要使用的源标签

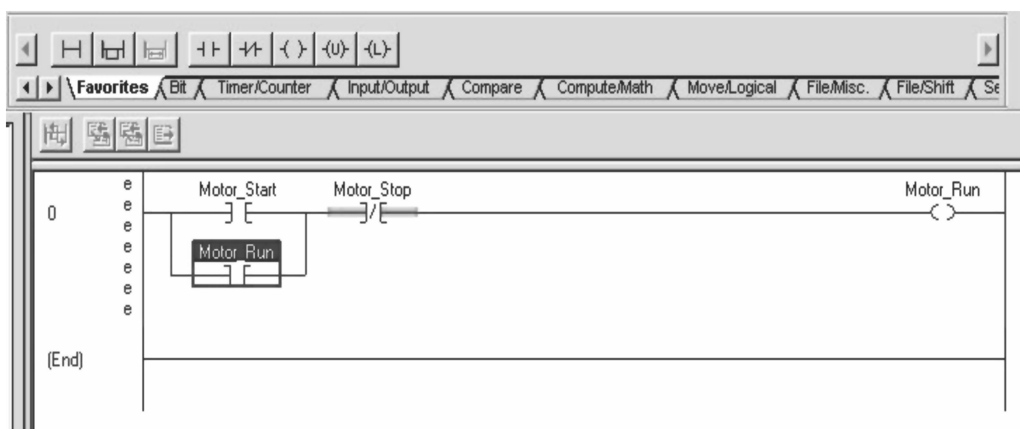


图 1-30 将源标签拖放至目标位置

14. 单击“(结束)”((End))梯级。现在这些“e”已消失。当单击鼠标取消编辑模式时，RSLogix 5000 软件将自动验证每个梯级。这使得编程更加容易。现在显示的梯级应与图 1-31 相同。

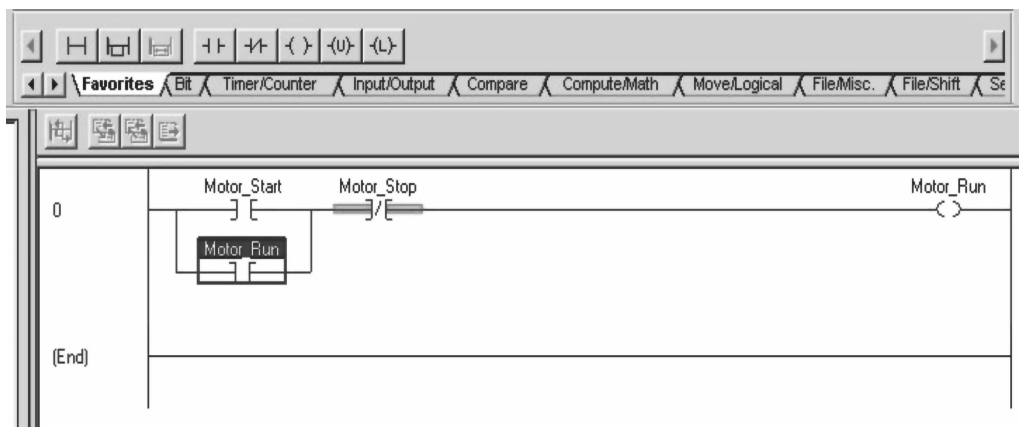


图 1-31 结束编辑后的程序梯级图

15. 在工具栏上单击“保存”(Save) 图标保存程序。

实验 2：组态 I/O

一、实验目的

现在将进行项目的 I/O 组态。要与 I/O 模块通信，必须将模块添加到 I/O Configuration 文件夹中。在此离线实验过程中，将显示为 ControlLogix 演示添加 1756 I/O，以及为每个 CompactLogix 演示添加 1769 I/O，因为该 I/O 是 1768-L43 和 1769-L35E 处理器的本机设备。按照适用于实验站上设备的步骤进行。将为显示一些高级 I/O 功能，例如：

1) 无负载检测；
2) 开路检测；
3) 输出验证；
4) 模拟量标定；
5) 模拟量报警。

将继续使用已打开的项目。在本实验中，将 ControlLogix I/O 添加到应用中。

二、添加 ControlLogix I/O

1. 在控制器项目管理器中，右键单击 1756 Backplane, 1756-A10 并选择“新建模块”(New Module)。

“选择模块”(Select Module) 窗口出现，如图 2-1 所示。

2. 选择“数字量”(Digital)，展开选项。

3. 选择 1756-IB16D 模块，如图 2-2 所示。

4. 单击“确定”(OK)。软件将要求选择模块的主版本，如图 2-3 所示。

5. 选择“主版本”(Major Revision) 2 并单击“确定”(OK)。将出现 1756-IB16D 的“模块属性”(Module Properties) 向导。

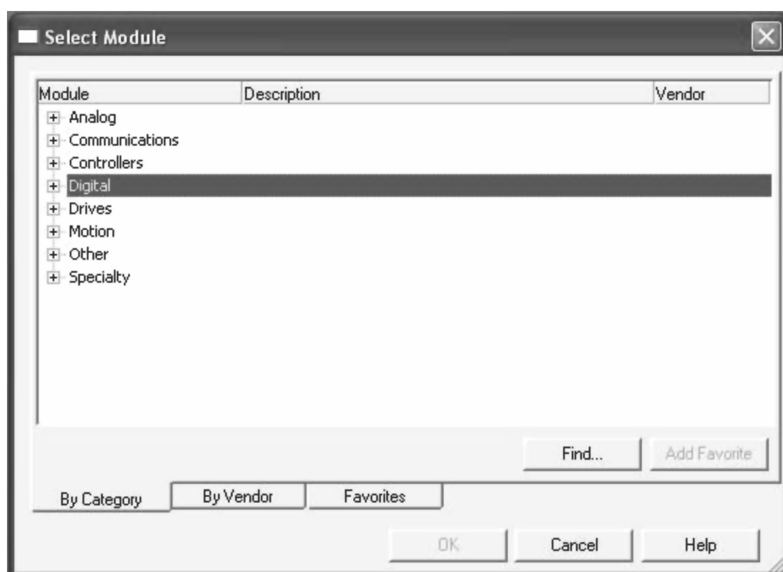


图 2-1 打开的模块选择列表窗口

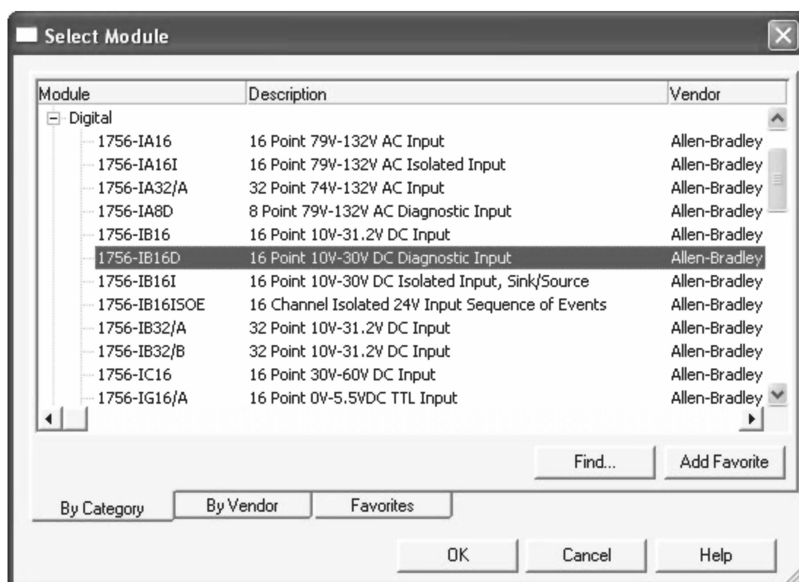


图 2-2 展开的数字模块列表

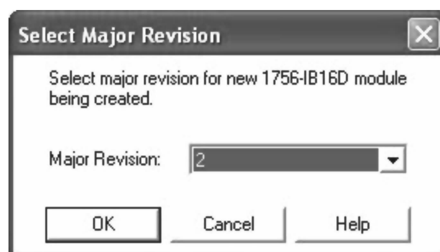


图 2-3 模块版本选择对话框

参考信息

模块组态向导

只要向系统中添加 I/O 模块就需要通过模块组态向导。该向导允许逐步通过某个模块所需的整个组态。稍后可以通过在 I/O Configuration 文件夹中双击相应模块或通过标签监视器/编辑器访问此信息。

在 Logix 中，组态 I/O 模块不再需要 DIP 开关或跳线。I/O 模块均通过软件组态。在进行系统设置时可节省时间。

所有模块的组态是控制器程序的一部分，而且这些组态可从控制器下载到模块中；从而能够在 I/O 模块失效时进行轻松替换。

6. 输入“名称”（Name）、“插槽”（Slot）和“电子匹配”（Electronic Keying）参数，如图 2-4 所示。保留所有其他字段的默认值。

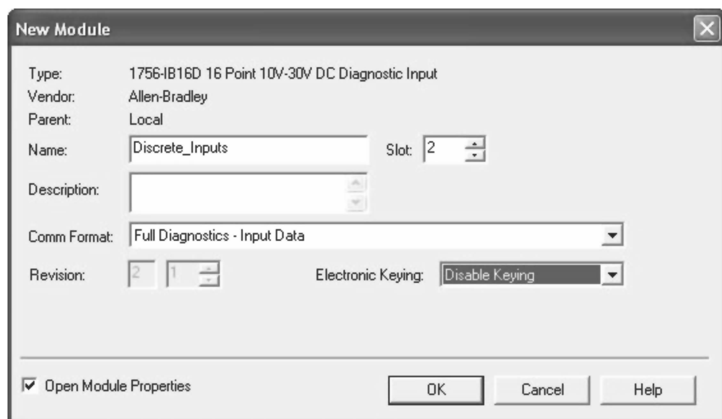


图 2-4　打开的数字量模块属性窗口

参考信息

通信格式（Comm Format）：

确定与模块关联的标签的数据结构。许多 I/O 模块支持不同的格式。每种格式使用不同的数据结构。

电子匹配（Electronic Keying）：

在将模块插入机架插槽时，控制器会从新插入模块读取的信息与用户在其项目中对该特定插槽的组态进行比较。将读取和比较以下数据：

1）供应商、产品类型、产品目录号、主版本、次版本。

2）在初始模块组态过程中，用户可在以下模块匹配选项中选择一个：

a）精确匹配（Exact Match）：上述参数必须全部匹配，否则插入的模块将拒绝连接。

b）兼容匹配（Compatible Module）：必须满足以下条件，否则插入的模块将拒绝连接：模块类型、产品目录号和主版本必须匹配，物理模块的次版本必须等于或大于软件中指定的版本。

c）禁用电子匹配功能（Disable Keying）：根本不使用电子匹配功能。

7. 单击“确定”（OK）以显示如图 2-5 所示的“连接属性页面”（Connection Property Page）。

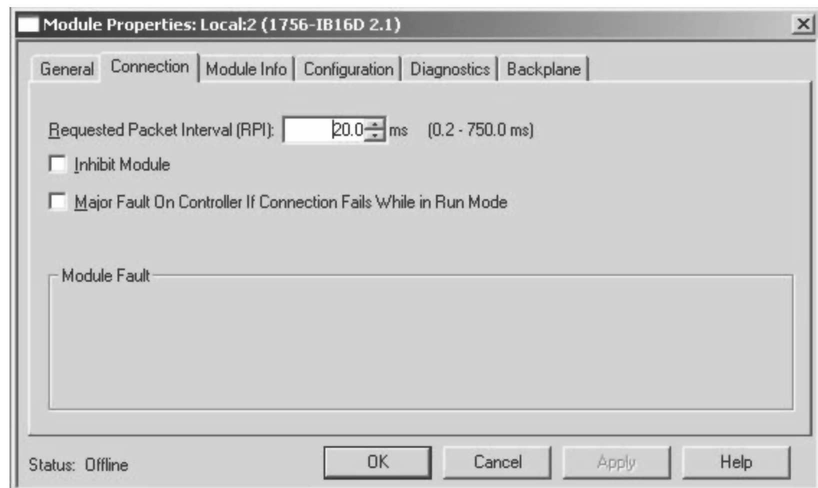


图 2-5 打开的数字量模块连接属性窗口

参考信息

请求信息包间隔（RPI）：

“请求信息包间隔”（Requested Packet Interval）指定向模块和从模块更新数据的周期。RPI 的组态单位为毫秒。范围为 0.2 ~ 750ms。ControlLogix 和 1768-L43 处理器允许分别组态各个 RPI 值，而 1769-L35e CompactLogix 处理器将 I/O 模块的连接视为经机架优化的连接，这意味着 1769 I/O 模块必须共享相同的 RPI。

8. 所查看的前两个界面是所有 I/O 模块通用的。从这里开始，其余界面则根据所选具体模块的不同而不同。

9. 选择“组态”（Configuration）选项卡。

将看到如图 2-6 所示界面。其中显示的是 1756-IB16D 模块特定的组态参数。

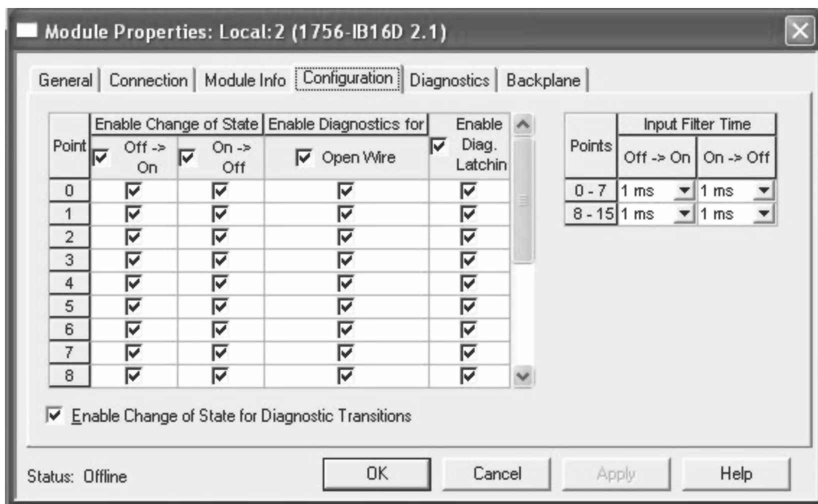


图 2-6 打开的数字量模块组态参数选项卡

10. 单击“确定”（OK）关闭向导。

在控制器项目管理器中，I/O Configuration 文件夹的显示应如图 2-7 所示。

11. 在控制器项目管理器中，右键单击 1756 Backplane, 1756-A10，并选择“新建模块”（New Module）。将出现“选择模块”（Select Module）窗口，如图 2-8 所示。

12. 选择“数字量”（Digital），展开选项。

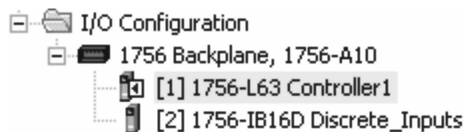


图 2-7 插入数字量模块后的项目列表

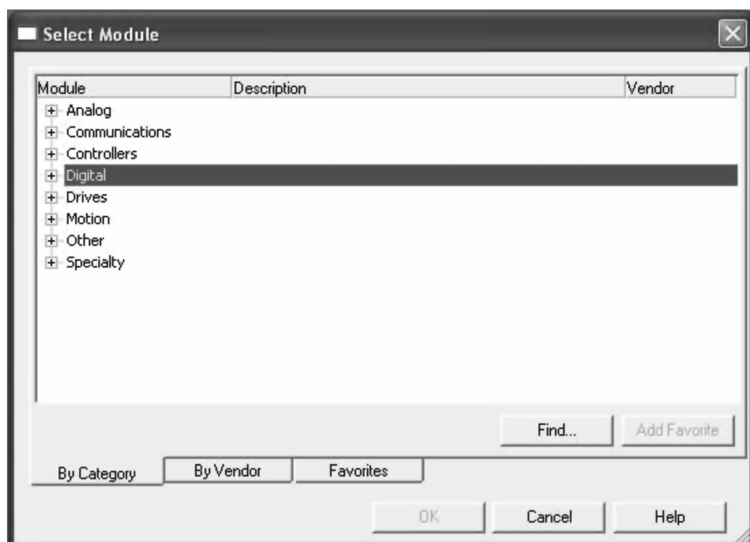


图 2-8 打开的模块选择列表窗口

13. 滚动浏览列表并找到 1756-OB16D，如图 2-9 所示。

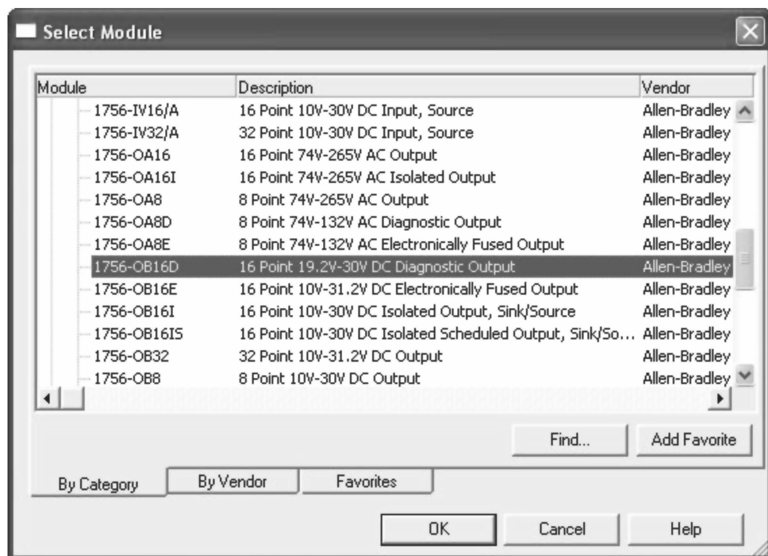


图 2-9 展开的数字模块列表模

14. 在列表中选择 1756-OB16D，然后单击“确定”（OK）。软件将要求选择模块的主版本，如图 2-10 所示。

15. 选择“主版本”（Major Revision）2，并单击“确定”（OK）。将出现 1756-OB16D 的“新建模块”（New Module）向导。

16. 输入“名称”（Name）、“插槽”（Slot）和“电子匹配”（Electronic Keying）参数，如图 2-11 所示。保留所有其他字段的默认值。

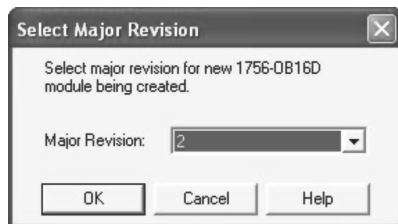


图 2-10 主版本选择对话框

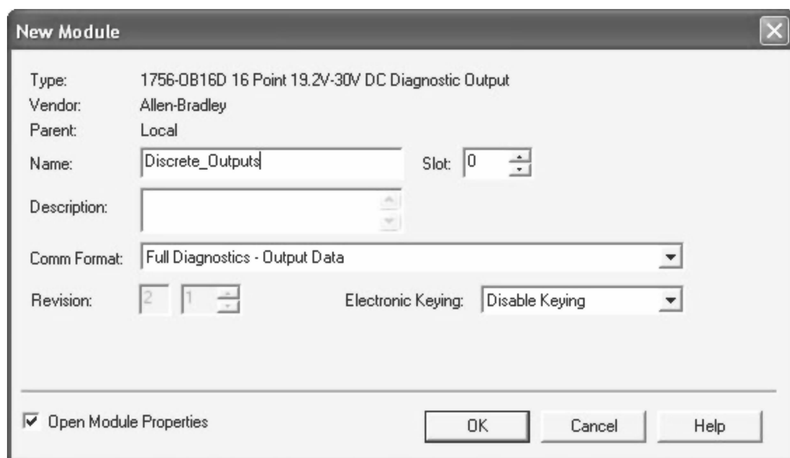


图 2-11 打开的数字量模块属性窗口

17. 单击“确定”（OK）。

18. 在“模块属性”（Module Properties）页面中选择“组态”（Configuration）选项卡。出现的组态数据是 1756-OB16D 模块所特定的，如图 2-12 所示。

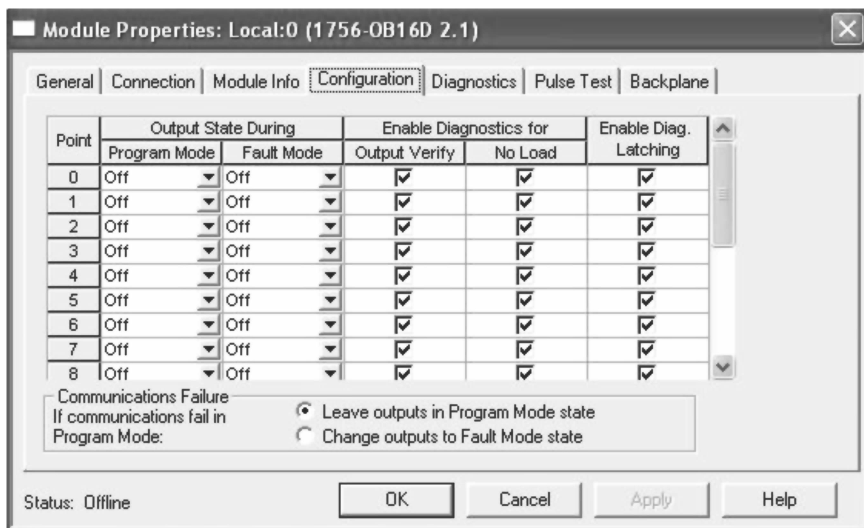


图 2-12 打开的数字量模块组态参数选项卡

19. 单击“确定”（OK），关闭向导。

20. 在控制器项目管理器中，验证 I/O Configuration 文件夹的显示，如图 2-13 所示。

21. 在控制器项目管理器中，右键单击 1756 Backplane, 1756-A10，并选择“新建模块”（New Module）。将出现“选择模块”（Select Module）窗口。

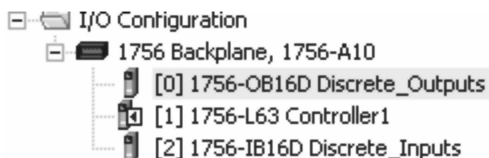


图 2-13 插入第二个数字量模块后的项目列表

22. 选择“模拟量”（Analog）。

23. 如前面一样地滚动浏览列表，找到并选择 1756-IF6I。

24. 单击“确定”（OK）。将出现 1756-IF6I 的“新建模块”（New Module）向导。

25. 输入参数如图 2-14 所示。

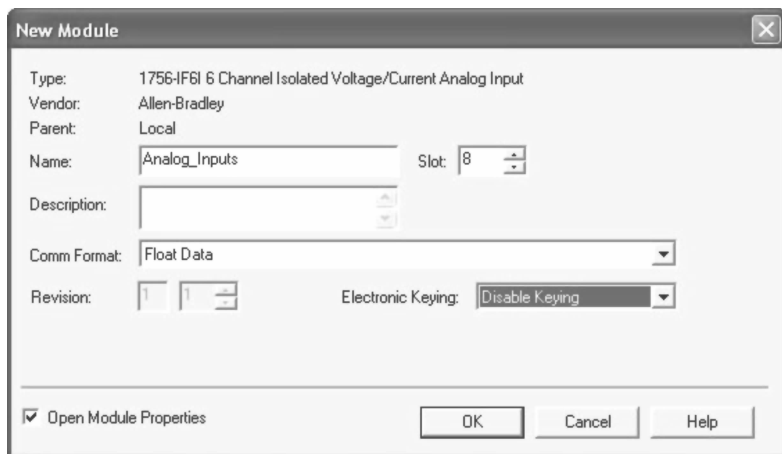


图 2-14 打开的模拟量模块属性窗口

26. 单击“确定”（OK）。在“模块属性”（Module Properties）页面中选择“组态”（Configuration）选项卡，如图 2-15 所示。出现的组态数据是 1756-IF6I 模块所特定的。

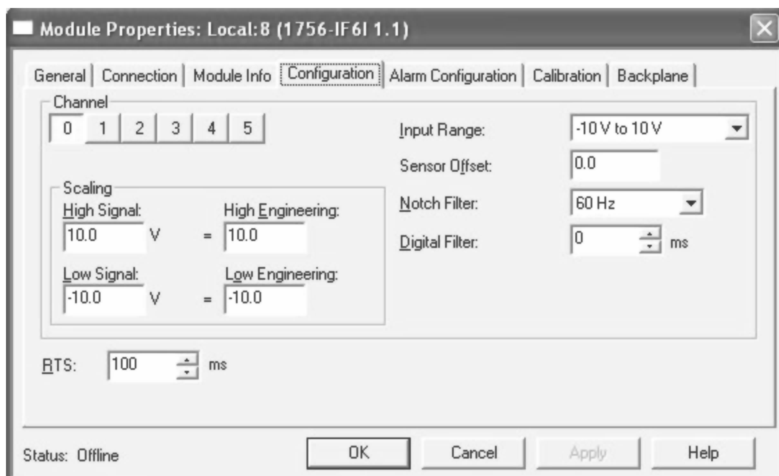


图 2-15 打开的模拟量模块组态参数选项卡

注意到了“标定”（scaling）部分。此模块确实板载有输入标定功能。与采用原始工程单位返回不同，可令模块返回采用诸如英寸、PSI、速度等工程单位的数据。这可节省应用的编程时间。

27. 单击“报警组态”（Alarm Configuration）选项卡，如图 2-16 所示。

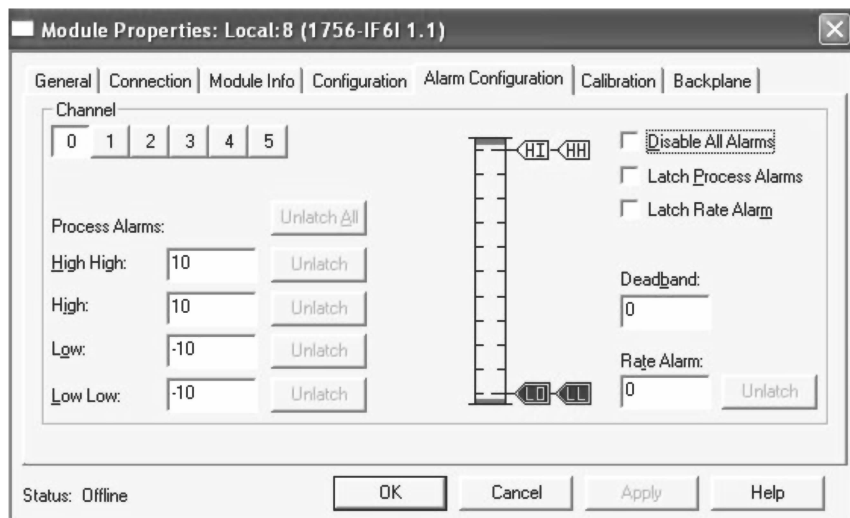


图 2-16 打开的模拟量报警组态参数选项卡

现在可看到，可对模块中的模拟量输入进行所需的报警操作。由于此操作板载于模块中，又可节省编程的时间。

28. 继续单击其他“模块属性”（Module Properties）选项卡，查看模拟量输入的各个选项。

29. 完成后单击“确定”（OK）。在控制器项目管理器中，I/O Configuration 文件夹的显示应如图 2-17 所示。



图 2-17 插入模拟量模块后的项目列表

30. 在工具栏上单击“保存”（Save）图标保存程序。

实验 3：将计算机连接到控制器

一、实验目的

在本实验中，将介绍使用 RSLogix5000 软件完成的在线操作。

1）启动 RSLinx 通信软件；

2）组态通信驱动程序。

二、启动 RSLinx 软件

在这部分实验中，将启动 RSLinx 软件，并通过该软件组态用来与演示箱中的 Logix 处理器通信的驱动程序。

1. 双击桌面上的 RSLinx 图标，启动 RSLinx 软件。

2. 单击 RSWho 图标。

将出现罗克韦尔软件 RSLinx Gateway-[RSWho-1]界面，如图 3-1 所示。

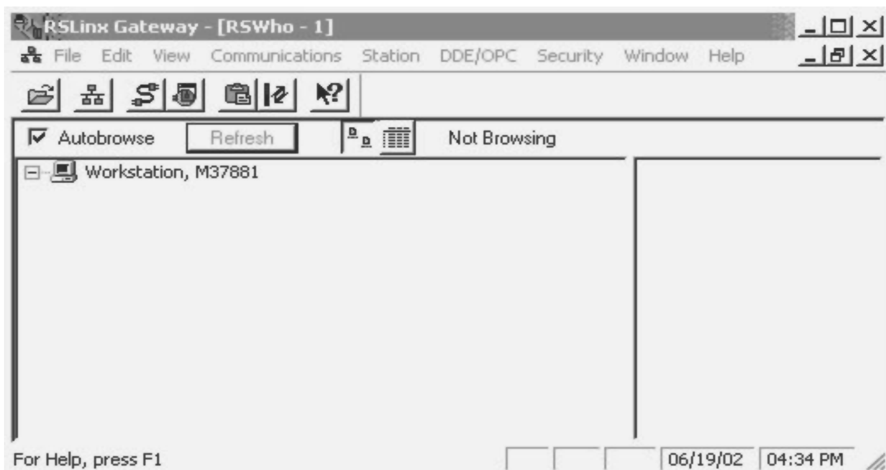


图 3-1　RSWhog 界面

参考信息

RSWho

RSWho 界面实际上是 RSLinx 的网络浏览器界面，借此可查看所有活动的网络连接。

此界面的左侧窗格是树形控件，该部分以层次视图显示网络和设备。当网络或设备已折叠，即指示为 + 符号时，可单击 + 符号或双击该网络或设备图标来展开视图并开始浏览。当网络或设备已展开，即指示为 - 符号时，可单击 - 符号或双击该网络或设备图标来折叠该视图。

RSWho 界面的右侧窗格是列表控件，该控件以图形化方式表示网络中存在的所有设备。

三、添加 AB_ETHIP-1（Ethernet/IP） 驱动程序

在这部分实验中，将添加用来与 Logix 处理器通信的 Ethernet/IP 驱动程序。

1. 在“通信”（Communications） 菜单中，选择“组态驱动程序”（Configure Drivers）。将显示“组态驱动程序”（Configure Drivers） 对话框，如图 3-2 所示。

2. 在“可用驱动程序类型”（Available Driver Types） 下拉菜单中，选择“EtherNet/IP 驱动程序”（EtherNet/IP Driver），然后单击“添加新驱动程序”（Add New） 按钮。

在 RSLinx 中，将注意到共列出了两种不同的以太网驱动程序：EtherNet/IP 驱动程序和以太网设备。一般情况下，应使用新的 EtherNet/IP 驱动程序 ... 它将自动扫描并查找网络中所有兼容 EtherNet/IP 的设备。但使用此驱动程序无法找到少数几种较早的罗克韦尔以太网产品。较早的以太网设备驱动程序适用于所有罗克韦尔以太网产品，但其仅会扫描手动令其搜索的 IP 地址。如果需要，可在 RSLinx 中同时激活两种不同类型的驱动程序和/或每种类型的多个实例。

3. 单击“确定”（OK） 接受默认名称（AB_ETHIP-1），如图 3-3 所示。

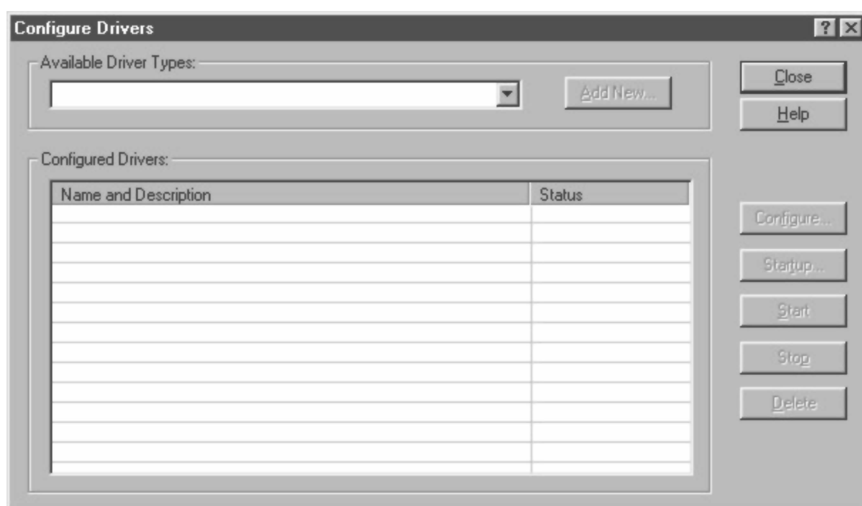


图 3-2 Configure Drivers 对话框

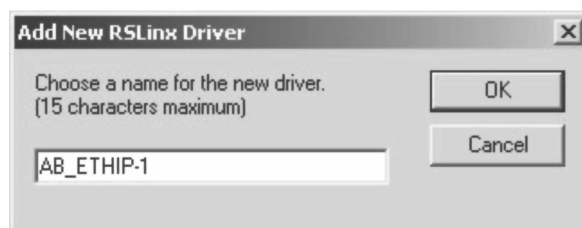


图 3-3 选择名字

4. 确保已启用“浏览本地子网” (Browse Local Subnet) 单选按钮，然后单击“确定”(OK)，如图 3-4 所示。

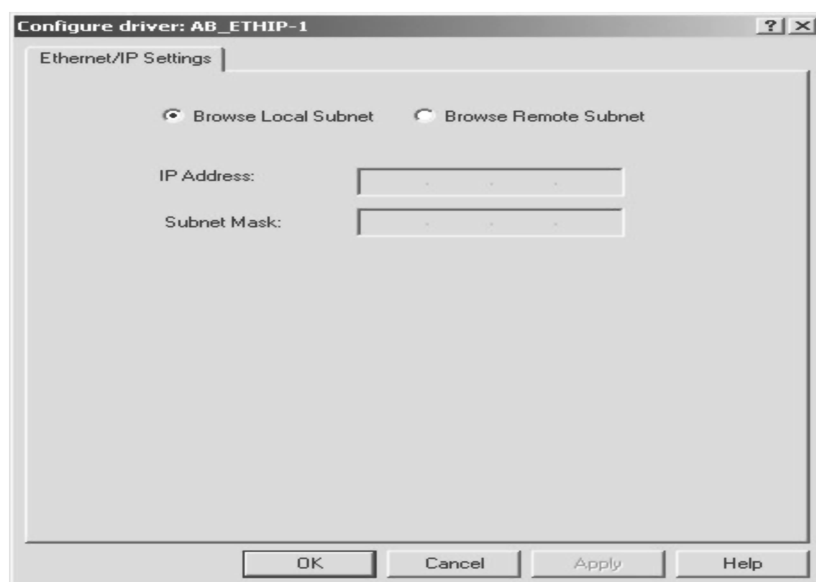


图 3-4 浏览本地子网

5. 通过单击“关闭”(Close)，退出“组态驱动程序”(Configure Driver) 对话框。

实验 4：将计算机中的项目加载到控制器

一、实验目的

1) 确定要使用的控制器的类型；
2) 打开与要使用的控制器对应的项目；
3) 将程序下载到控制器；
4) 将使用实验 1 所执行的步骤中创建的程序。

二、将项目下载到控制器

在这部分实验中，将下载项目。

1. 最大化 RSLogix 5000 和 Controller1. ACD 项目。

2. 在“通信”(Communications) 菜单中，选择“在线节点”(Who Active)，如图 4-1 所示。

将出现“在线节点”(Who Active) 界面，如图 4-2 所示。

3. 通过单击“+”展开视图，直到显示如图 4-3 所示界面。

4. 单击“下载”(Download)。软件将要求验证该下载。

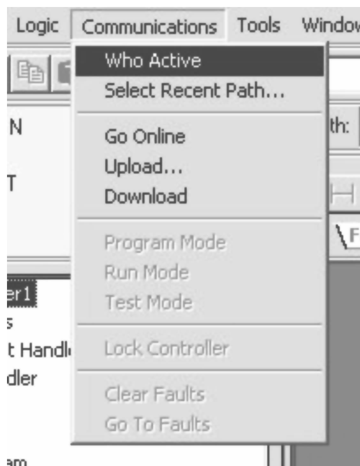


图 4-1 通信 (Communications) 菜单

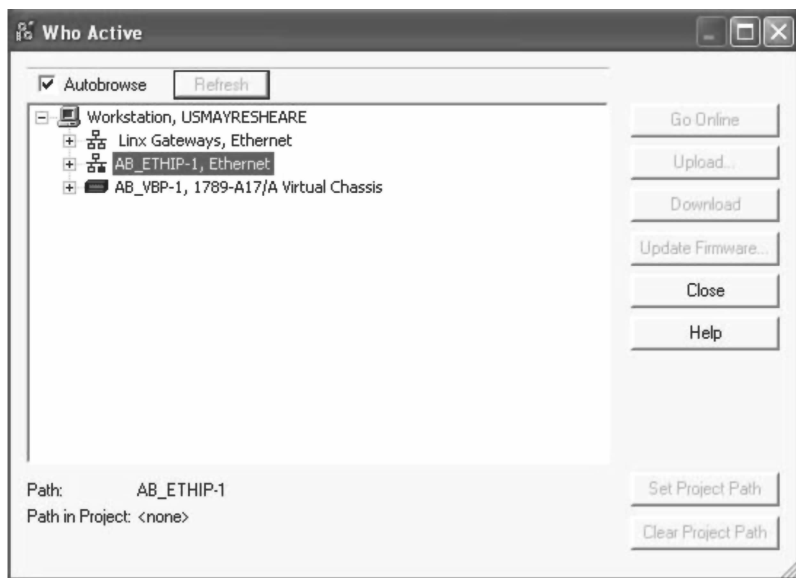


图 4-2 在线节点 (Who Active) 界面

5. 出现以下揭示后，再次单击“下载”(Download)，如图 4-4 所示。

6. 然后便会开始将项目下载到控制器。

如果控制器在下载前处于“运行”模式，软件将提示 返回到“运行”模式。如果要求，则选择“是”(YES)。

此时将与控制器在线连接，状态 LED 将模拟控制器上的 LED。

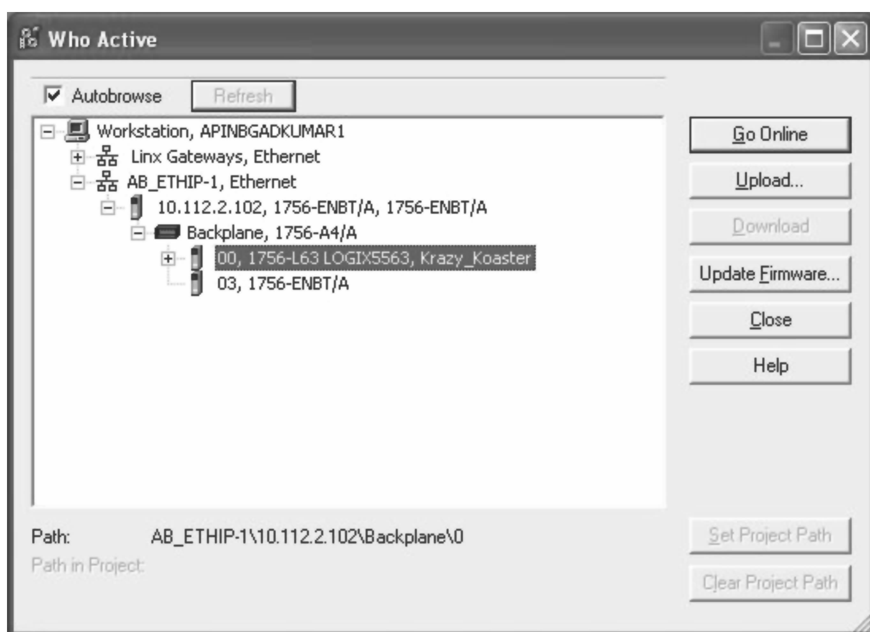


图 4-3 在线节点列表

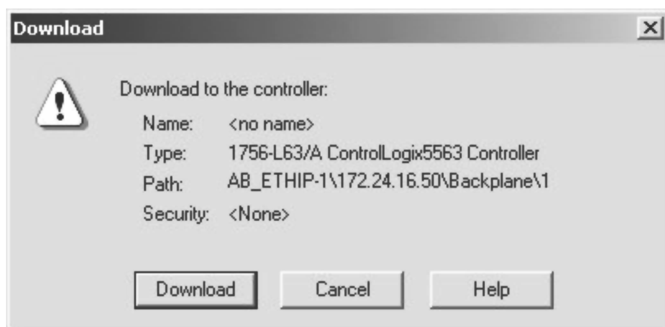


图 4-4 确认下载对话框

实验 5：使用 RSLogix 5000 帮助演示

一、实验目的

在本实验中，将了解功能强大的 RSLogix 5000 在线帮助系统。在本实验中，将查看：

1） 指令帮助；

2） 模块接线图；

3） 在线参考资料；

4） 第三方供应商示例项目；

5） 起始页-快速入门。

二、指令帮助

1. 在“帮助”（Help）下拉菜单中选择“指令帮助”（Instruction Help），如图 5-1 所示。

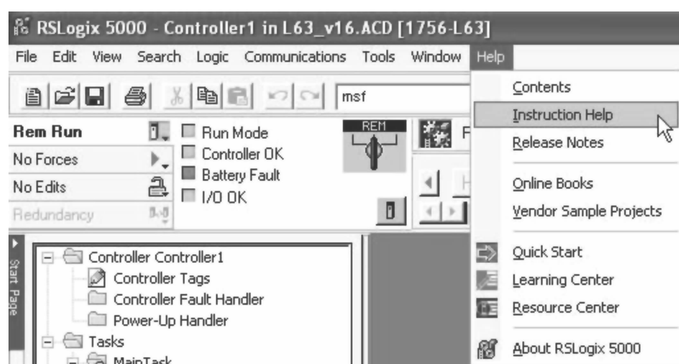


图 5-1 指令帮助菜单

将出现以下窗口，如图 5-2 所示。

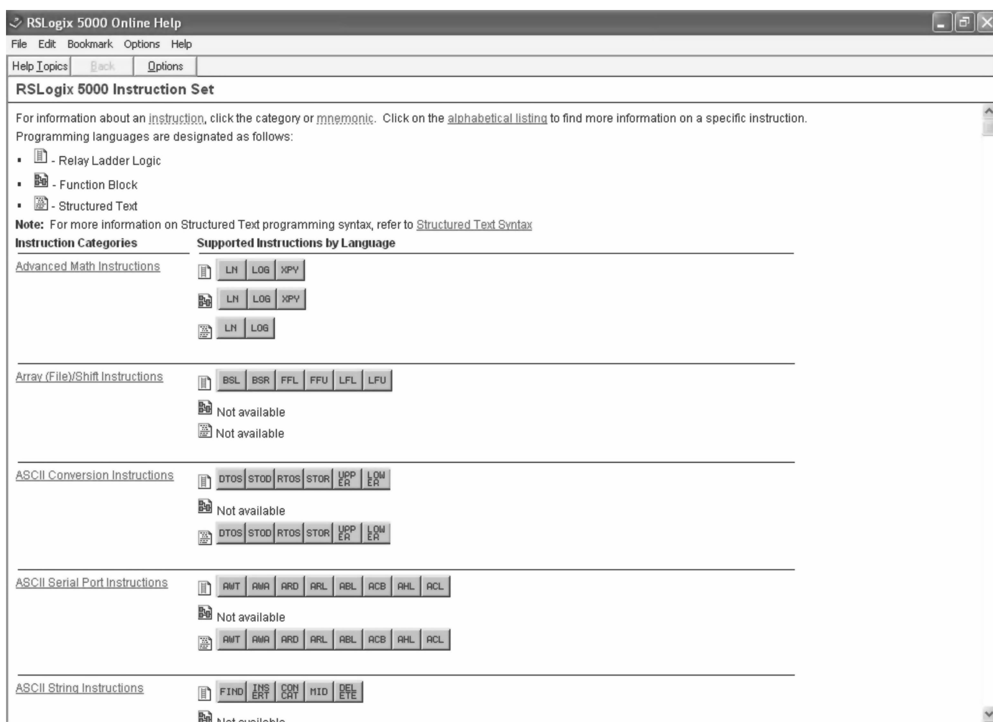


图 5-2 指令帮助窗口

2. 单击指令找到其说明、其参数的详细信息、相关指令，以及有关如何使用该指令的示例。

三、查看 I/O 模块接线图

1. 在“帮助主题”（Help Topics）下拉菜单中选择“目录”（Contents）。
2. 如果尚未选择“查找”（Find）选项卡，则选择该选项卡。
3. 如图 5-3 所示填写字段 1。
4. 如图 5-3 所示在字段 2 中选择 IA16。

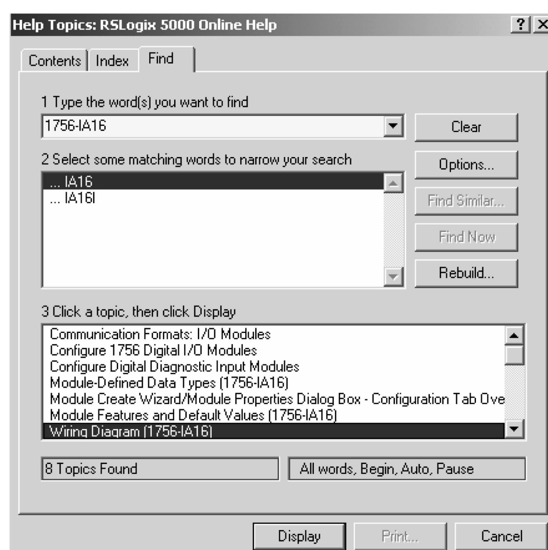


图 5-3 填写标．查找的实际字段

5. 在字段 3 中，向下滚动列表并找到“接线图（1756-IA16）”（Wiring Diagram（1756-IA16））。

6. 单击“显示”（Display）查看该模块的接线图，如图 5-4 所示。请注意，可能需要将界面最大化。

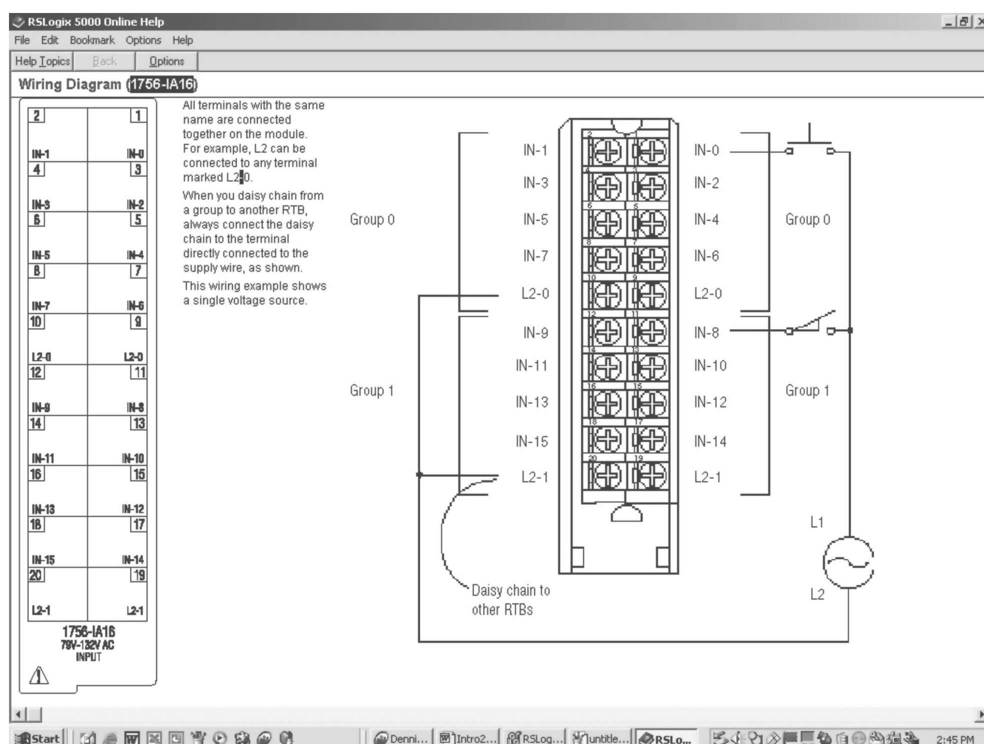


图 5-4 模块的接线图

7. 接线图查看完毕后，关闭显示窗口。

四、使用在线参考资料

1. 在“帮助”（Help）下拉菜单中选择“在线书籍”（Online Books），如图 5-5 所示。

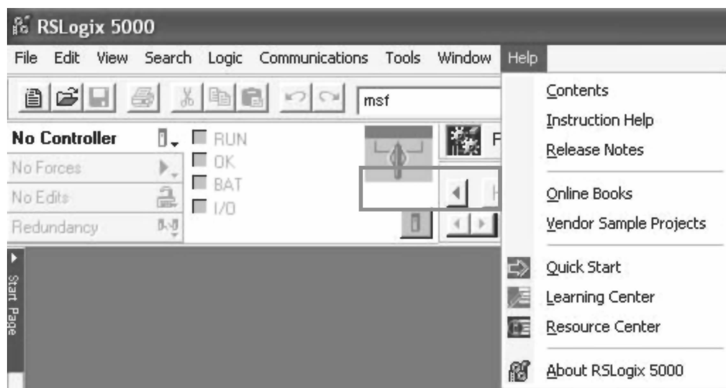


图 5-5 在线书籍选项

安装 RSLogix 5000 软件时，还可选择安装在线参考资料。如果已安装这些资料，则可在这里通过 RSLogix 5000 帮助查看。

2. 通过此工具浏览可用的各类型手册。

五、第三方供应商示例项目

1. 在“帮助”（Help）下拉菜单中选择“供应商示例项目”（Vendor Sample Projects）。Adobe Acrobat 查看器将打开并显示多个第三方模块，从而允许查看并重复使用 RSLogix 5000 的示例程序，如图 5-6 所示。

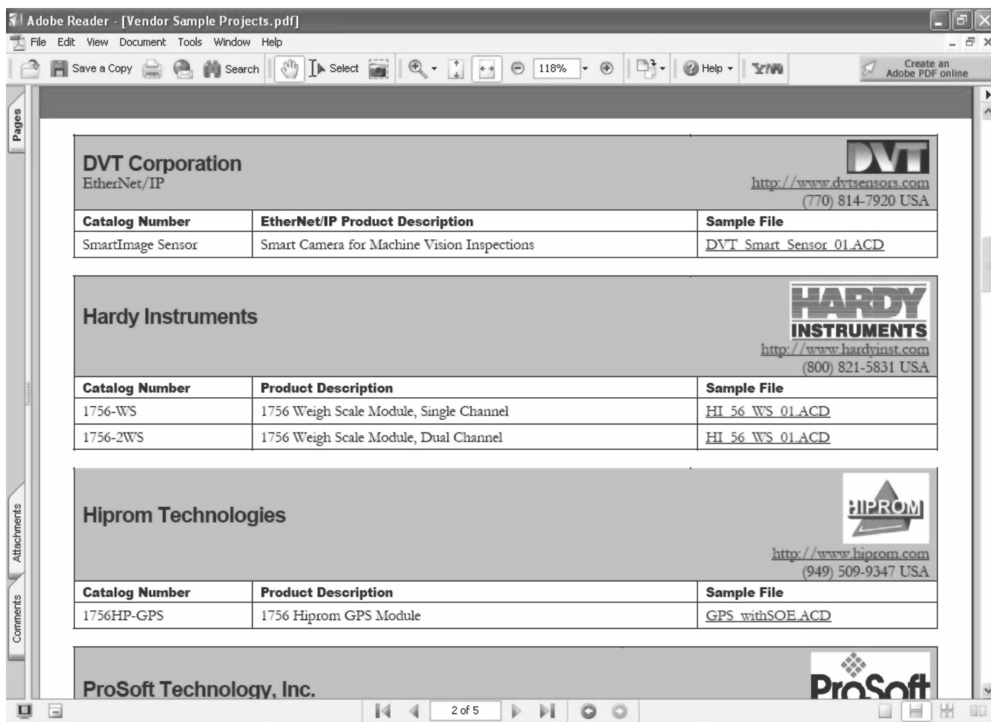


图 5-6 供应商示例项目

2. 完成后关闭 Adobe Acrobat。

六、快速入门页面

在“帮助”（Help）下拉菜单中选择“快速入门”（Quick Start）。单击“快速入门”（Quick Start）查看其中内容，如图 5-7 所示。

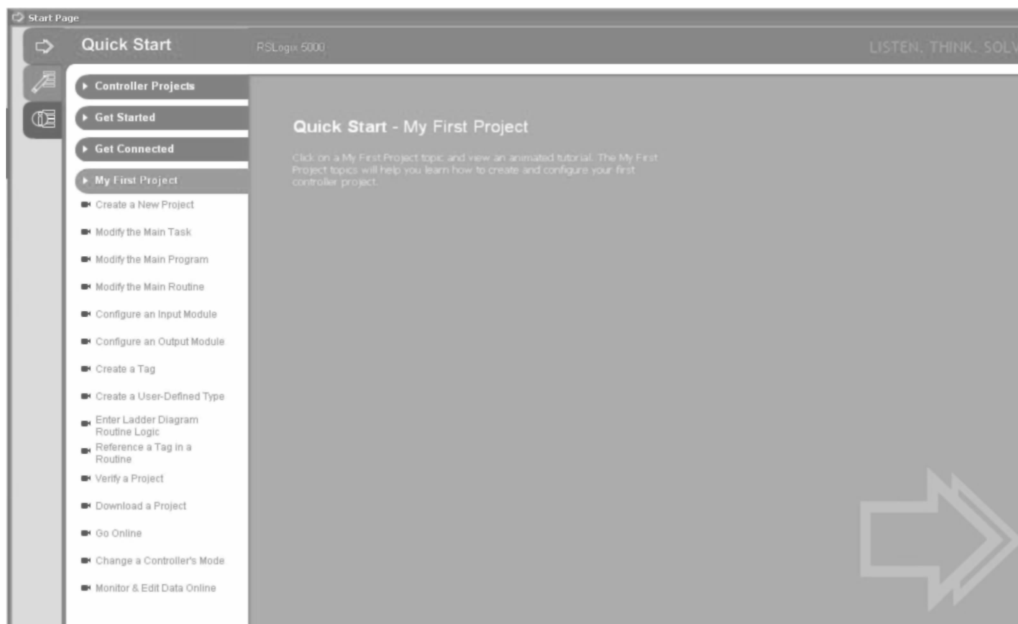


图 5-7　快速入门界面

实验 6：标签和数据范围划定

一、实验目的

在这部分实验中，将查看 RSLogix 5000 中的标签监视器/编辑器。还将讨论控制器范围标签与程序范围标签的概念。

二、将继续使用该已打开的项目

1. 在控制器项目管理器中，双击“控制器标签”（Controller Tags），步骤如图 6-1 所示。

将出现标签监视器/编辑器窗口。可看到窗口左下角的两个选项卡，它们分别标记为“监视标签”（Monitor Tags）和“编辑标签”（Edit Tags），如图 6-2 所示。

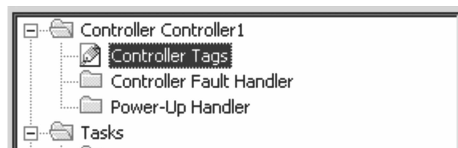


图 6-1　选择“控制器标签”

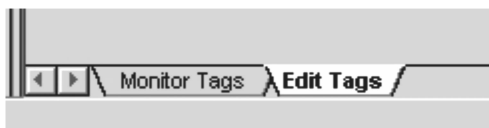


图 6-2　监视器/编辑器窗口

参考信息

监视/编辑标签选项卡

选择“监视标签”（Monitor Tags）选项卡后，将显示标签的实际值。例如，如果要查

看输入按钮，则软件将显示该按钮已主动激活或禁用。

选择“编辑标签”（Edit Tags）选项卡后，可创建新标签或修改现有标签属性。

如果无法创建或修改标签参数，请验证已选择“编辑标签”（Edit Tags）选项卡。

首先会注意到其中存在 I/O 模块标签，然后会想起来这是前面刚创建的 3 个标签，如图 6-3 所示。



图 6-3　标签编辑器窗口

注意“标签编辑器”（Tag Editor）窗口左上角处标记为“范围”（Scope）的字段。在本实验的前面部分，已简要地介绍过控制器范围和程序范围的标签。当前的选择是 Controller1（控制器），如图 6-4 所示。



图 6-4　标签“范围”选择窗口

之前创建标签时，是在程序范围中创建的。

2. 单击“范围”（Scope）选择框的向下箭头。

3. 选择“程序→主程序”（Programs→MainProgram），如图 6-5 所示。

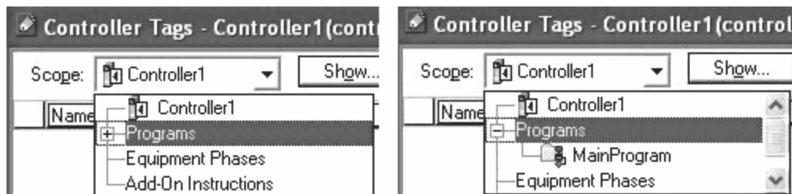


图 6-5　控制程序选择列表

标签编辑器的视图现已切换到程序级别，可查看之前创建的标签，如图 6-6 所示。

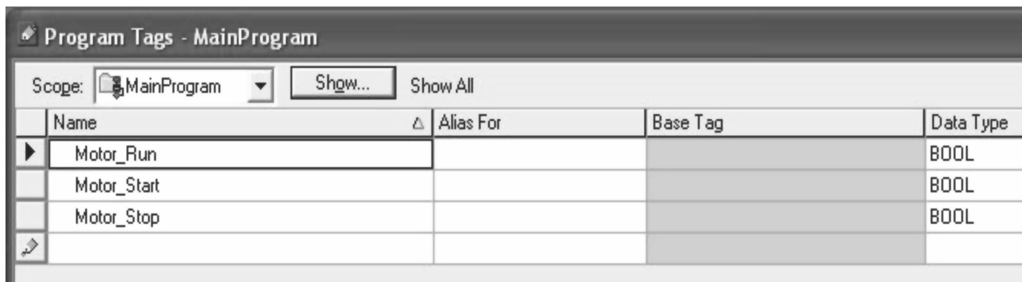


图 6-6　将已有标签导入编辑器

4. 在工具栏上单击“保存”（Save）图标，保存程序。

三、查看 ControlLogix I/O 标签

已在项目中组态了 ControlLogix I/O 模块；现在看一下在 RSLogix 5000 中是如何显示该信息的。

将继续使用该已打开的项目。

1. 在控制器项目管理器中，双击“控制器标签”（Controller Tags）。

2. 将出现标签编辑器窗口，如图 6-7 所示。向右拖动以增加标签名称字段的宽度。这样便可查看整个标签名称。

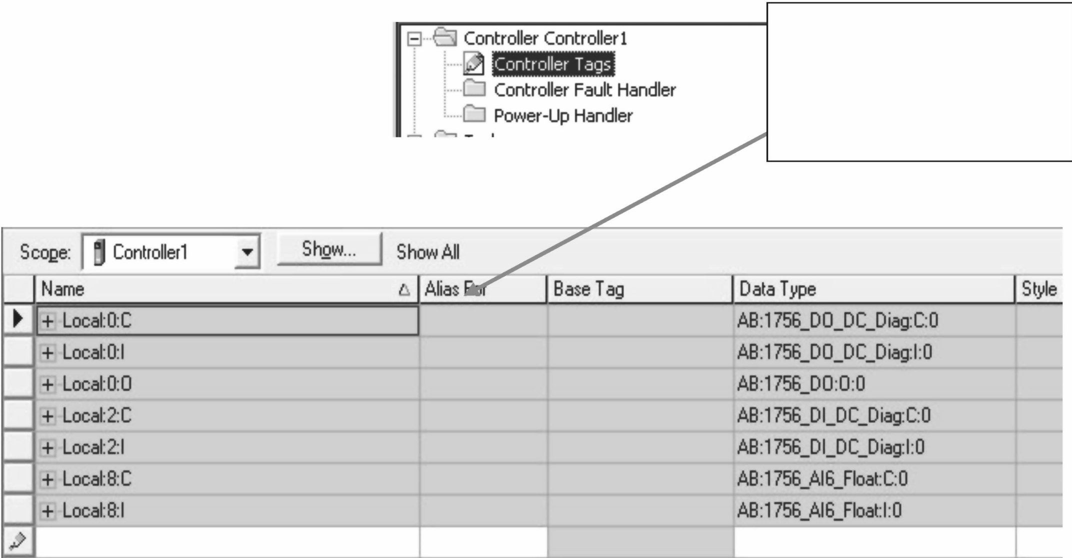


图 6-7 标签编辑器窗口

通过查看标签编辑器的左上角，可注意到当前处于控制器范围内。所有 I/O 模块标签均是在控制器范围内创建的，如图 6-8 所示。

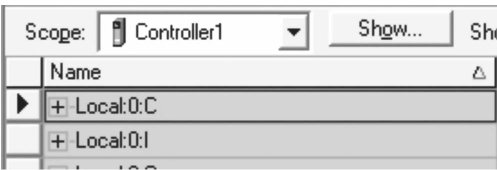


图 6-8 控制器范围选择界面

3. 通过单击“监视标签”（Monitor Tags）选项卡切换到“监视标签”（Monitor Tags），如图 6-9 所示。

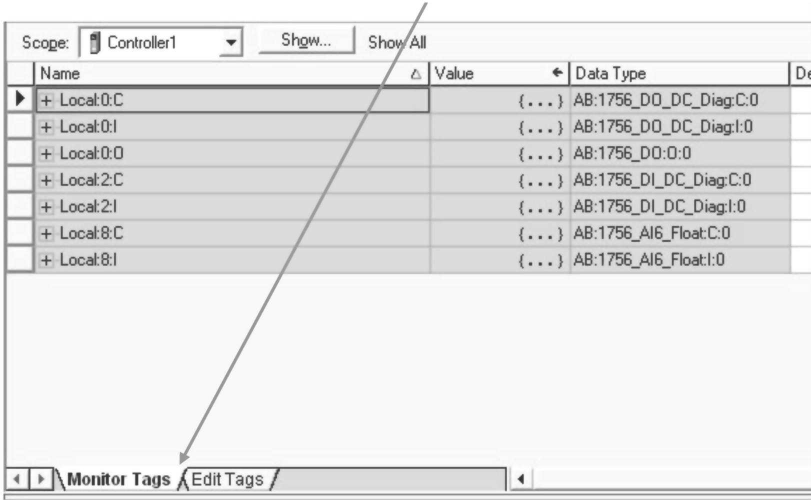


图 6-9 “监视标签”（Monitor Tags）界面

上面的条目是已添加模块的标签结构。其中包含的标签要多于实际显示的标签。注意标签名称旁边的 + 符号，这表示可展开标签结构以查看更多信息。

4. 通过单击 +，展开并查看 I/O 模块的标签。

对于各个模块，“组态”（Configuration）标签下将显示在模块组态向导中输入并选择的所有数据。

在查看 I/O 模块的标签时，可看到输出模块 1756-OB16D 具有关联的输入数据。可能会问，为什么输出模块会有输入？OB16D 是诊断模块。所有诊断数据都通过这些输入返回到控制器。

5. 在工具栏上单击“保存”（Save）图标保存程序。

四、分配别名标签

在这部分实验中，将了解别名标签。将继续使用该已打开的项目。

注：确保所使用的是离线文件。

参考信息

别名

别名标签允许用户创建代表另一个标签的标签：

1）两个标签共享相同的值；

2）当一个标签值改变时，另一个标签也反映该变动。

在下列情况下使用别名：

1）在绘制接线图前对逻辑进行编程；

2）为 I/O 设备分配描述性名称；

3）为复杂标签提供更简单的名称；

4）为数组元素使用描述性名称。

1. 在控制器项目管理器中，双击“主例程”（MainRoutine），如图 6-10 所示。

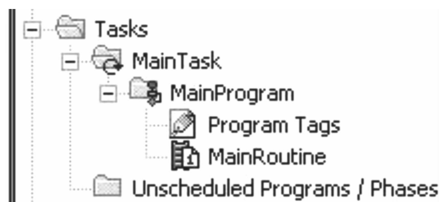


图 6-10　项目管理器列表

将出现梯形图编辑器，如图 6-11 所示。

在实验的上一部分中，已向项目中添加了 I/O 模块。现在来为 I/O 模块的程序中的标签指定别名。

1）将 Motor_Start 指定为插槽 2 中 1756-IB16D 的输入点 0 的别名。

2）将 Motor_Stop 指定为插槽 2 中 1756-IB16D 的输入点 1 的别名。

3）将 Motor_Run 指定为插槽 0 中 1756-OB16D 的输出点 0 的别名。

2. 右键单击标签 Motor_Start，并选择“编辑‘Motor_Start’属性”（Edit‘Motor_Start’Properties），如图 6-12 所示。

将出现 Motor_Start 的“标签属性”（Tags Properties）窗口。当前该标签定义为基础标签。

3. 选择“别名”（Alias）作为“类型”（Type），并注意“标签属性”（Tag Properties）窗口会发生变动，如图 6-13 所示。

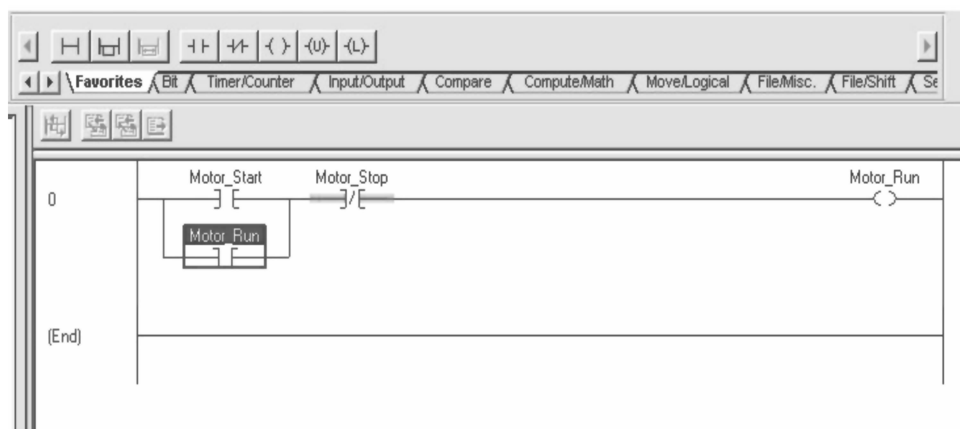


图 6-11　梯形图编辑器界面

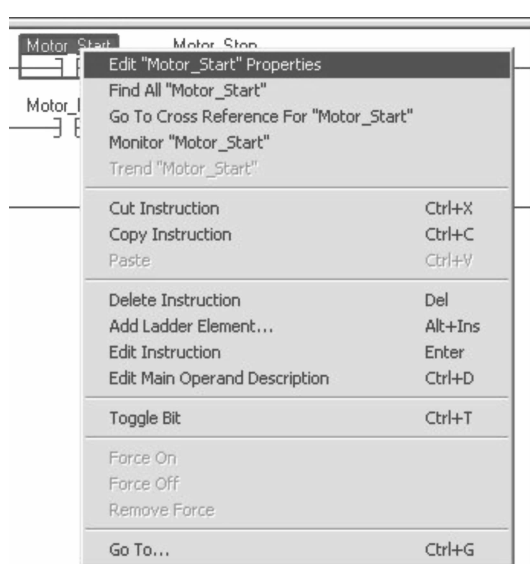


图 6-12　选择“编辑项目”条目

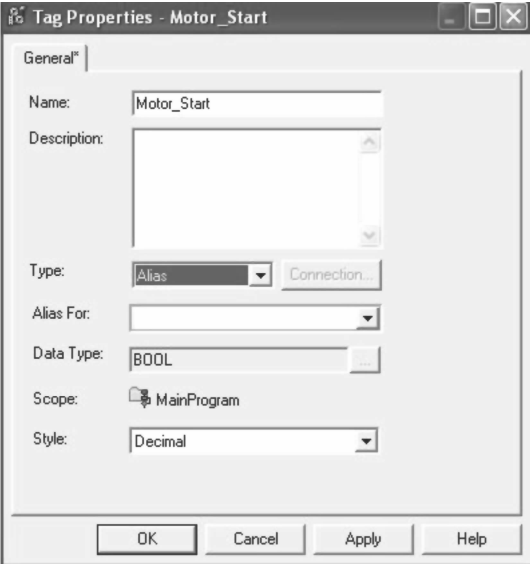


图 6-13　打开的“标签属性”窗口

4. 单击“别名”（Alias For）的向下箭头。将出现标签浏览器，如图 6-14 所示。

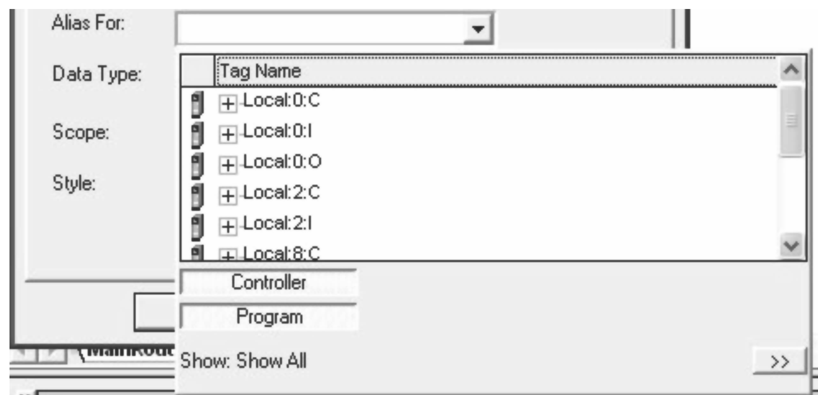


图 6-14　“别名”标签浏览器

在浏览器中同时显示控制器范围和程序范围的标签。需要在控制器范围标签中选择地址。

5. 单击“程序范围标签”（Program Scoped Tags）按钮取消选择“程序范围标签”（Program Scope Tags）。

界面将更改为仅显示“控制器范围标签”（Controller Scoped Tags）（注：此窗口可调整大小），如图 6-15 所示。

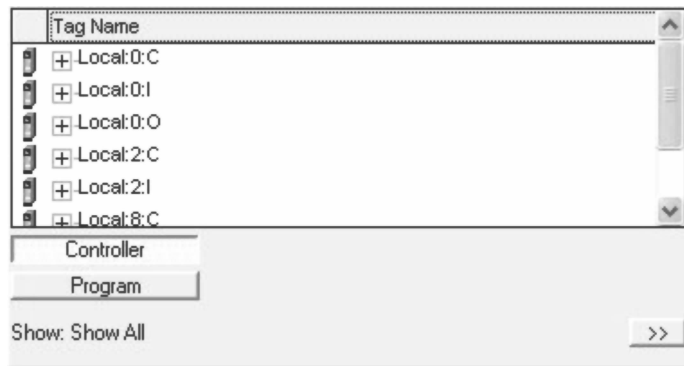


图 6-15 控制器范围标签

注：此窗口可调整大小。

6. 展开 Local:2:I 并选择 Local:2:I. Data。

7. 单击 Local:2:I. Data 数据类型（DINT）旁边的向下箭头。这将打开 1756-IB16D 模块的数据点表格。

8. 选择表格中的 0，如图 6-16 所示。

当选择标签浏览器中的 0 后，该窗口将关闭，并且将出现“标签属性”（Tag Properties）。对话框，如图 6-17 所示。

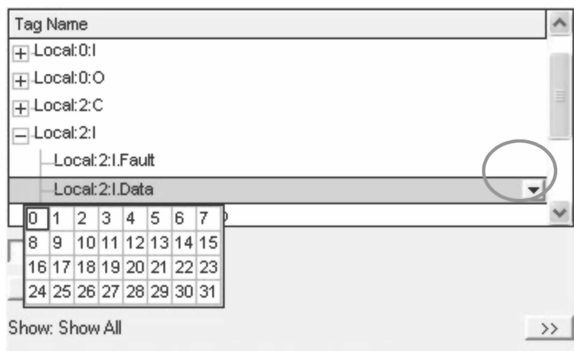


图 6-16 选中数据表中的“0”位

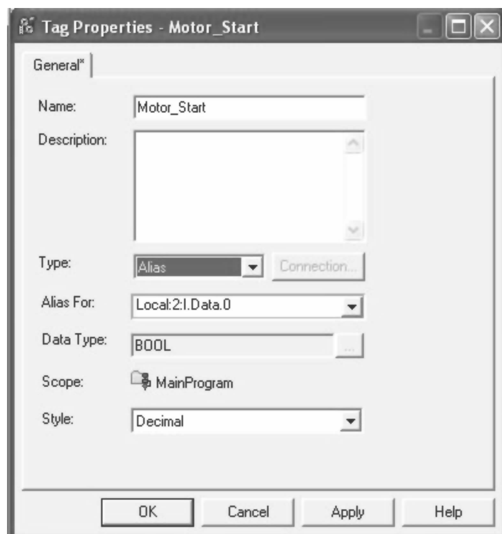


图 6-17 完成别名制定

Motor_Start 现在已被指定为 Local:2:I. Data. 0 的别名。这表示插槽 2 中的 1756-IB16D。

9. 单击“确定”（OK），关闭并将所做更改应用到标签 Motor_Start；如图 6-18 所示。

> 仔细查看梯形图代码中的 Motor_Start 标签。在标签 Motor_Start 下可看到 <Local:2:I. Data. 0>。这表示 Motor_Start 已被指定为 Local:2:I. Data. 0 的别名。这意味着标签在代码中彼此所代表的意义相同。Motor_Start 要比 Local:2:I. Data. 0 更易于阅读。

10. 使用以前的步骤，指定剩余两个标签的别名如下：

1）Motor_Stop = Local:2:I. Data. 1

2）Motor_Run = Local:0:O. Data. 0

11. 完成后，梯形图代码应如图 6-19 所示。

12. 在工具栏上单击“保存”（Save）图标保存程序。

13. 最小化 RSLogix 5000 软件。

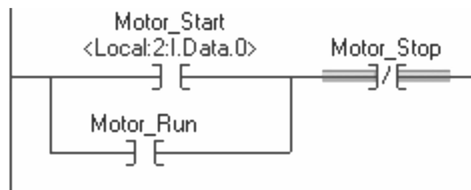


图　6-18

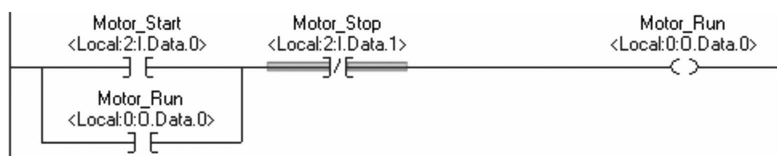


图　6-19

实验 7：测试逻辑程序

一、实验目的

在本实验中，将验证程序的操作情况。

注：在开始测试的逻辑前，请先下载程序。

> **参考信息**
>
> **I/O 映射**
>
> 演示箱中有一组针对此实验的按钮。这些按钮的映射关系如下：
>
> 1）Motor_Start = DI0
>
> 2）Motor_Stop = DI1
>
> 3）Motor_Run = DO0

二、将控制器置于运行模式并测试程序

1. 如果尚未处于运行模式，则在“控制器面板”（Controller Faceplate）中选择“运行模式”（Run Mode），如图 7-1 所示。

控制器将转到运行模式。这可通过查看控制器上的运行 LED 进行验证。现在该 LED 应发绿色光。也可在 RSLogix 5000 中通过查看控制器面板进行验证，如图 7-2 所示。

可注意到，这与控制器面板非常相似。

2. 在控制器项目管理器中单击 MainProgram 的“+”，以将其展开，如图 7-3 所示。

3. 双击 MainRoutine，以调出梯形图编辑器，如图 7-4 所示。

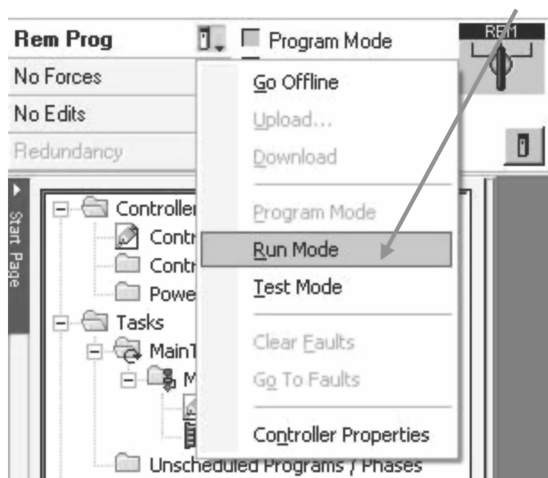


图 7-1 选中控制器面板“运行模式”

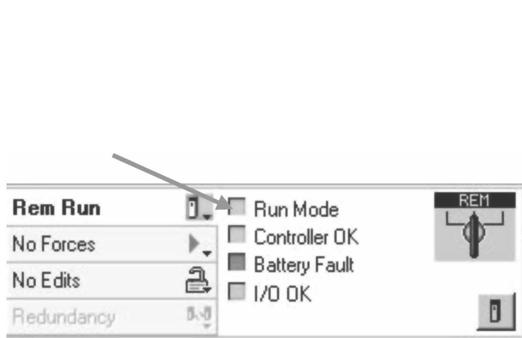


图 7-2 控制器面板 LED 指示

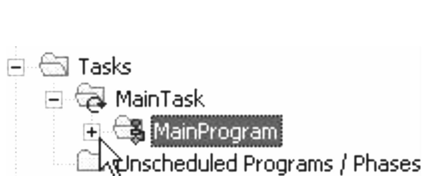


图 7-3 项目管理器列表

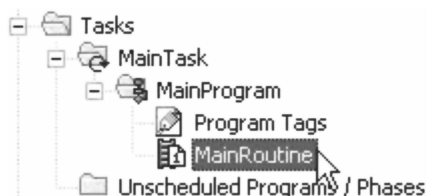


图 7-4 选中双击“MainRoutine”

重要事项!

在程序中，实际上将发现是之前在实验练习中组态的别名地址。出于本实验的考虑，已删除对别名的所有引用。保留的别名原样未动；如图 7-5 所示。

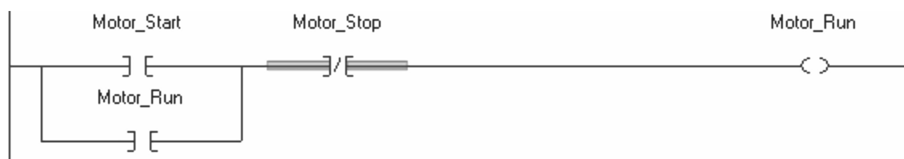


图 7-5 梯形图逻辑

现在将看到梯形图逻辑。注意梯形图两侧的绿色电源卡轨。这表示处于在线状态，且例程正在执行。

可注意到 XIO 指令 Motor_Stop 为绿色。这意味着该指令处于“真”（true）或“开启”（on）状态。这是因为 Motor_Stop 按钮未按下，如图 7-6 所示。

4. 按下 ControlLogix 按钮面板上的 DI1 按钮。这将会使 XIO 指令与 Motor_Stop 关联。可注意到绿色消失（见图 7-7）。这是因为该指令不再为真，如图 7-7 所示。

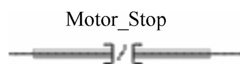


图 7-6 XIO 指令为真

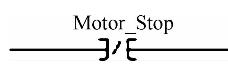


图 7-7 XIO 指令为假

5. 按下按钮 DI0（Motor_Start）。XIC 指令将变为真，并变成绿色。Motor_Run 将通电（变为绿色）。实验站上的指示灯 DO0 将亮起，如图 7-8 所示。

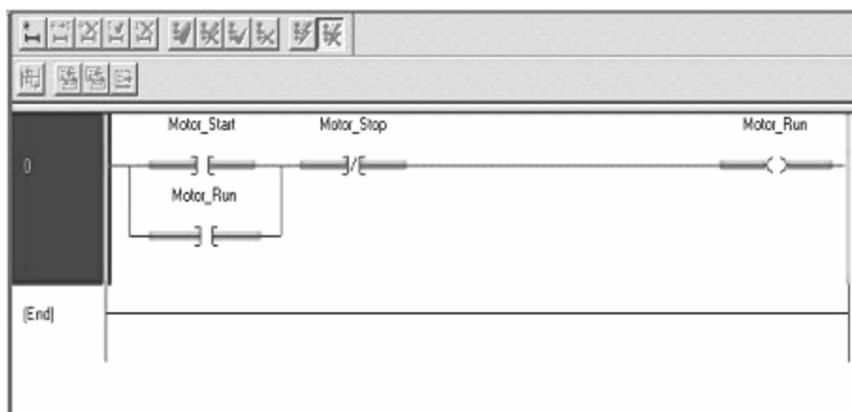


图 7-8　按下“Motor_Run”后的梯形图

6. 验证输出 DO0（Motor_Run），在释放按钮 DI0（Motor_Start）后仍为亮起状态。刚写入的梯形图逻辑是简单的三线控制或者电动机起动/停止制动电路，如图 7-9 所示。

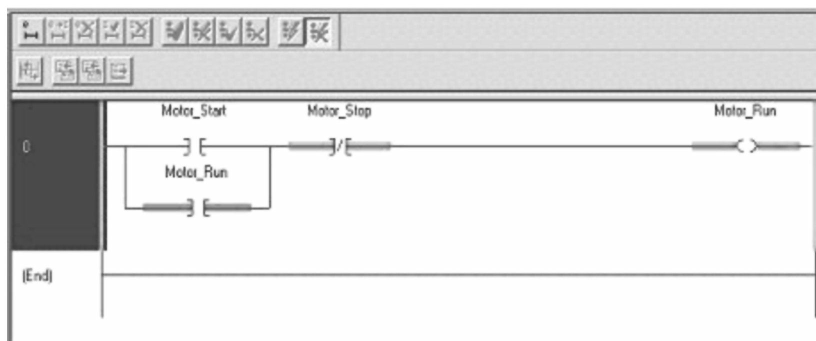


图 7-9　松开“Motor_Run”后的梯形图

7. 按下按钮 DI1（Motor_Stop），并验证输出 DO0（Motor_Run）已关闭，如图 7-10 所示。

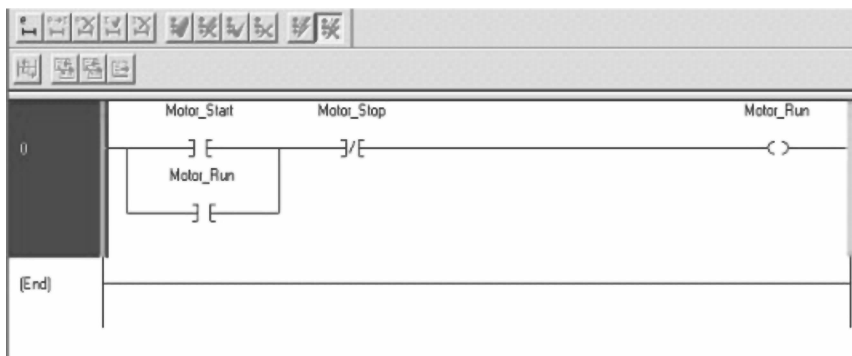


图 7-10　按下“Motor_Stop”后的梯形图

实验 8：在线添加逻辑和标签

一、实验目的

在本实验中，将执行在线编辑操作。

1）向逻辑中添加一个定时器，并使其基于电动机运行而执行；

2）添加梯形图逻辑，以在电动机停止时重置定时器。将继续使用该已打开的项目。

二、向逻辑中添加定时器

1. 右键单击梯级零左侧的蓝色区域，并选择“开始待决梯级编辑” （Start Pending Rung Edits），如图 8-1 所示。

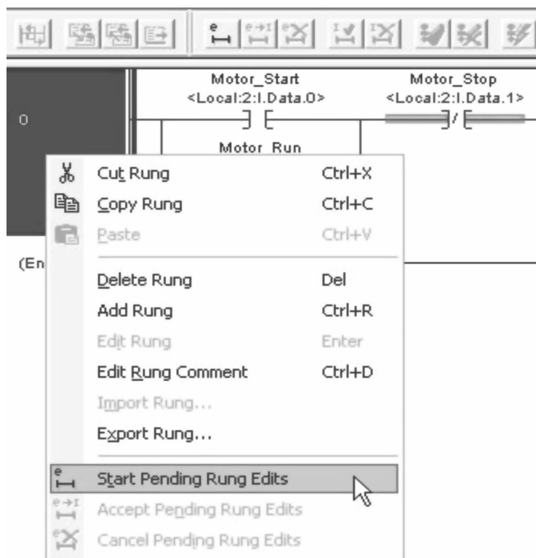


图 8-1　选择“Start Pending Rung Edits”

2. 现在的梯形图编辑器类似图 8-2 所示。

电源卡轨上带“I”的梯级是 将执行编辑的梯级。

3. 单击 OTE 指令，使其突出显示，如图 8-3 所示。

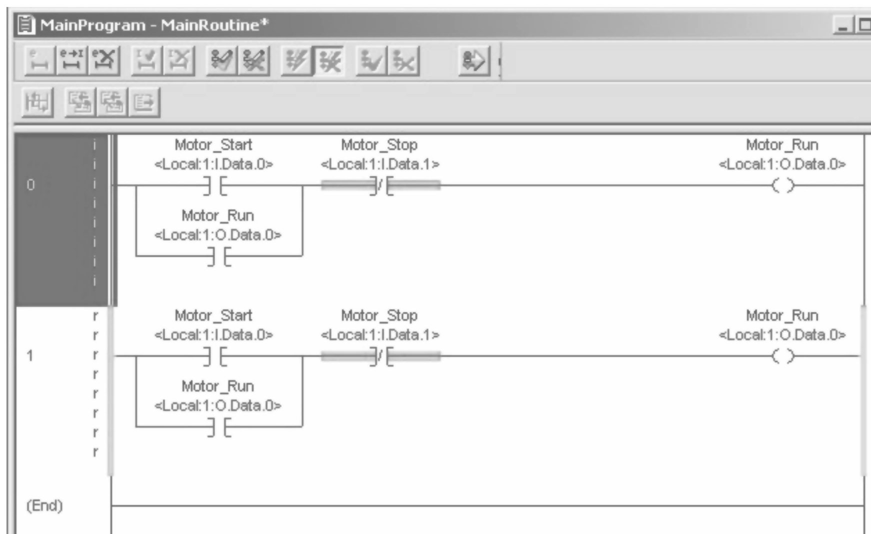


图 8-2　进入待决梯级编辑状态的梯形图

4. 在指令工具栏中单击“定时器/计数器”（Timer/Counter）选项卡，如图 8-4 所示。



图 8-3　选中 OTE 指令标签



图 8-4　选中“定时器/计数器”选项卡

5. 单击“定时器开启（TON）”（Timer On（TON））TON图标，如图 8-5 所示。

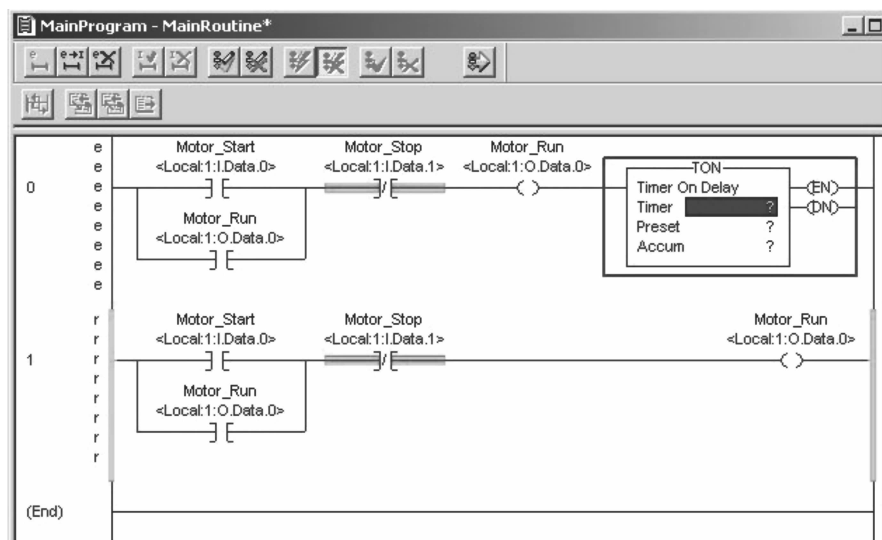


图 8-5　插入的定时器

已在 OTE 指令右侧的代码中插入一个定时器。

在 RSLogix 5000 中可将多个输出指令串在一起，不必创建分支。

6. 在定时器指令上，右键单击文字“定时器”（Timer）旁边的蓝色区域，并选择“新建标签”（New Tag），如图 8-6 所示。

将出现“新建标签”（New Tag）窗口。可注意到“数据类型”（Data Type）已设置为“定时器”（TIMER）。这是因为正在定时器指令中创建标签。

7. 在“名称”（Name）字段中输入“Timer”，然后单击“确定”（OK），如图 8-7 所示。

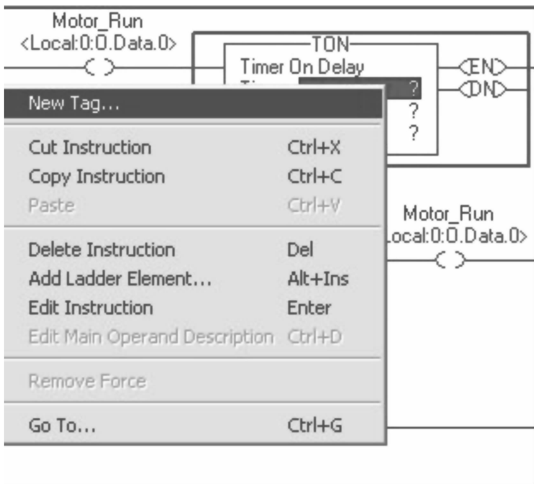


图 8-6　添加定时器标签

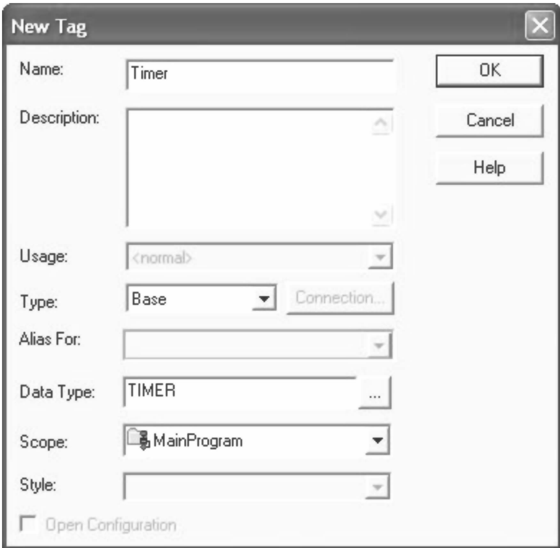


图 8-7　输入“定时器”标签信息

8. 验证已在定时器指令中创建的标签，如图 8-8 所示。

9. 在定时器指令中双击文字“预设值”（Preset）旁边的 0。

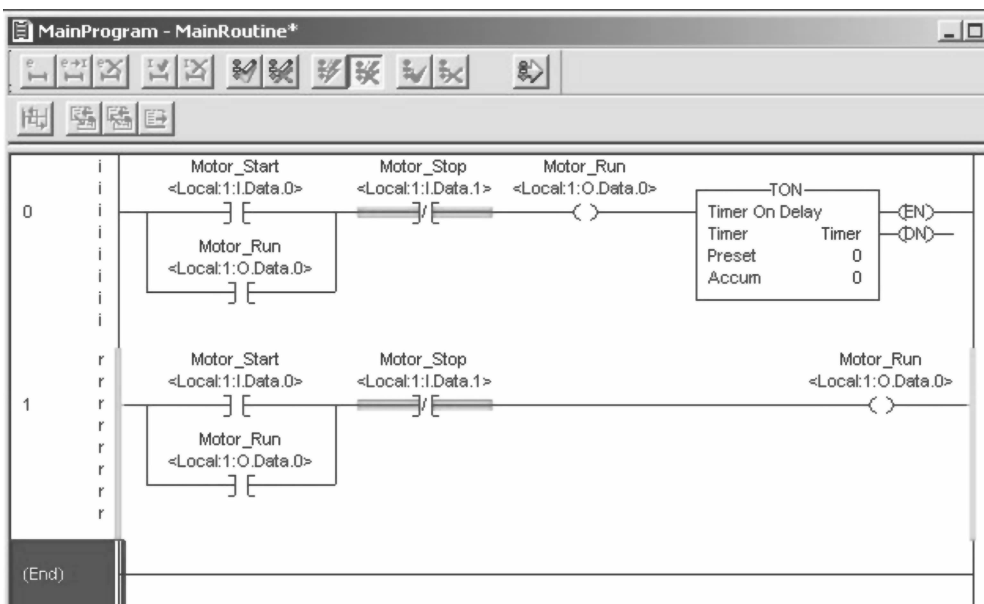


图 8-8 创建标签后的定时器

10. 输入值 32767。在 Logix 中，“定时器预设值”（Timer Preset）为 32 位 DINT，这意味着定时器的最大值为 2,147,483,647。

11. 按回车键。现在显示的 TON 指令应与图 8-9 所示相同的“预设值”（Preset）现在为 32767ms。保留累计值为零。现在已准备好验证所进行的编辑。

12. 单击“确定所有编辑”（Finalize All Edits）图标。

13. 在询问是否确定所有编辑时，单击“是”（YES）。梯形图编辑器现在显示如图 8-10 所示。

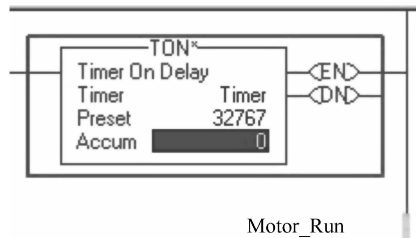


图 8-9 配置定时器参数

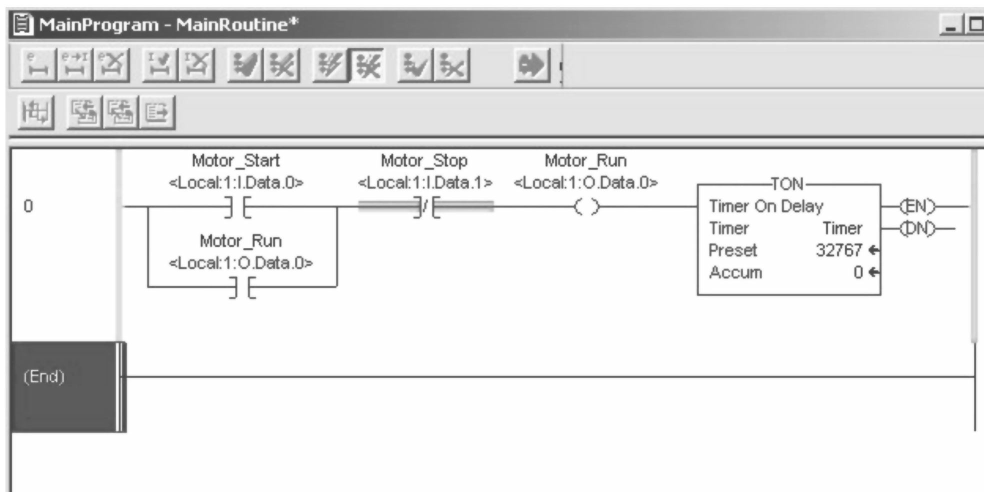


图 8-10 “确定所有编辑”后的梯形图

三、测试逻辑

1. 按下 DI0（Motor_Start）按钮。
2. 验证 DO0（Motor_Run）已亮起，定时器指令已开始递增。
3. 现在按下按钮 DI1（Motor_Stop）。
4. 验证 DO0 已关闭，定时器已重置。
5. 保存并退出 RSLogix 5000。

实验 9：规划 Logix 控制器中的标签数据库

一、实验目的

在本实验中，将介绍并演示如何通过用户自定义数据类型和数据范围划定来规划标签数据库。

1）了解使用 UDT 的优势；

2）学习如何优化 UDT 规划；

3）使用数据范围划定帮助简化，并加快开发工作。

现在将重点关注 Logix 控制器中的数据规划。

二、打开现有控制器文件

在实验的本部分中，将打开一个现有程序。

1. 在计算机桌面上，双击 Lab Files 文件夹。
2. 双击名为 Conveyor_Program_S3. ACD 的现有项目，将在 RSLogix 5000 中启动该项目。

三、为传送带创建用户自定义数据类型

已重新组织了程序规划以更好地利用 Logix，现在已准备好，并开始对数据规划进行重新组织。可注意到，工程师规划数据的方式仍像传统 PLC 使用带有整数、实数和定时器数据表一样。问题是，当与设备关联的数据分布到控制器内存各处时便很难进行跟踪。已再次决定充分利用 Logix，使用用户自定义数据类型。

> **参考信息**
>
> **用户自定义数据类型**
>
> 用户自定义数据类型也称为 UDT 或结构，借此按逻辑方式对数据进行组织或分组，以便所有与设备关联的数据都可组合在一起。
>
> 例如，每个传送带都有 8 个整数值、3 个实数值、2 个定时器和 11 个与其关联的布尔值。在传统的 PLC 中，可能需要 4 个不同的数据表。然后，当具有多条传送带时，可能需要详细地将传送带映射到各个数据表中。这样就会变得很难管理。
>
> 通过 UDT 能够实现的是将不同的数据类型（整数、实数、定时器、布尔等）组合到一起，共同作为用户自定义数据类型。然后便可创建该 UDT 类型的数组。这可使得编程工作、代码的记录和数据的跟踪都更加轻松。

1. 在控制器项目管理器中，双击“控制器标签”（Controller Tags），如图 9-1 所示。

将出现标签编辑器。可注意到，标签数据库的外观与传统 PLC 非常类似，如图 9-2 所示。

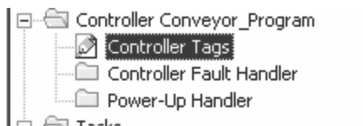


图 9-1　选中“控制器标签”

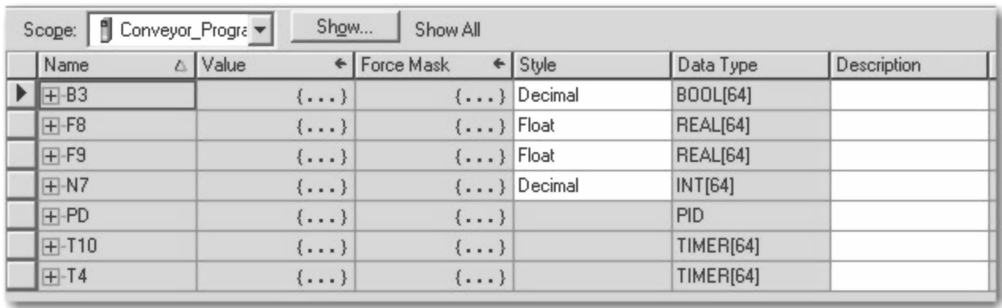


图 9-2 标签编辑器界面

参考信息

Logix 标签数据库

在传统 PLC 中，用物理地址标识各个数据项，例如 N7:0 或 B3:0/0。不使用地址注释或符号，就很难阅读逻辑。

在 Logix 控制器中没有固定的内存 N7:0 或 B3:0/0。Logix 内存是基于标签的系统。标签名称本身就可以标识数据。借此便能够组织数据以模拟机器。可在实际中使用分配的标签名称记录代码，例如 Nozzle_1_Temperature。

参考信息

创建 Logix 项目标签的指导原则见表 9-1。

表 9-1 指导原则

指导原则	详细信息
1）创建用户自定义数据类型	用户自定义数据类型(结构)是可以组织数据以匹配机器或进程。用户自定义数据类型提供以下优势： 一个标签包含系统特定方面的所有相关数据。这样将相关数据保存在一起便于查找，不考虑其数据类型 每个数据(成员)得到一个描述性名称。这样自动为逻辑创建初始水平的文档 可以使用该数据类型创建具有相同数据规划的多个标签 例如，使用用户自定义数据类型存储罐的所有参数，包括温度、压力、阀门位置和预设值。然后基于该数据类型为每个罐创建标签
2）使用数组快速创建一组类似标签	数组在一个公共标签名称下创建数据类型的多个实例 数组使可以组织一组使用相同数据类型并执行类似功能的标签 以一维、二维或三维组织数据以匹配数据表示的内容 例如，使用二维数组组织油库的数据。数组的每个元素表示一个油罐。元素在数组中的位置表示罐的地理位置 重要说明：尽量减少使用 BOOL 数组。许多数组指令不能在 BOOL 数组上使用。这使得更难以初始化和清除 BOOL 数据数组 通常对 PanelView 屏幕的位级别对象使用 BOOL 数组。否则，使用 DINT 标签的各个位或 DINT 数组

（续）

<table>
<tr><th>指导原则</th><th colspan="2">详细信息</th></tr>
<tr><td rowspan="6">3）利用程序范围标签</td><td colspan="2">如果希望多个标签具有相同名称，请在每个不同程序的程序范围定义每个标签（程序标签）。这是可以在多个程序中重新使用逻辑和标签名称
避免同时为控制器标签和程序标签使用相同的名称。在程序中，如果程序的程序标签与控制器标签名称相同，则不能引用控制器标签
某些标签必须是控制器范围（控制器标签）</td></tr>
<tr><td>如果希望将标签用于：</td><td>则分配此范围：</td></tr>
<tr><td>项目中的多个程序中</td><td rowspan="3">控制器范围（控制器标签）</td></tr>
<tr><td>消息（MSG）指令中</td></tr>
<tr><td>在控制器之间产生或使用数据与 PanelView 终端通信</td></tr>
<tr><td>以上都不是</td><td>程序范围（程序标签）</td></tr>
<tr><td>4）对于整数，使用 DINT 数据类型</td><td colspan="2">要提高逻辑效率，请尽可能少用 SINT 或 INT 数据类型。条件允许时，对整数使用 DINT 数据类型
Logix 5000 控制器通常将值作为 32 位值（DINT 或 REAL）进行比较或操作
控制器在使用值前，通常会将 SINT 或 INT 值转换为 DINT 或 REAL 值
如果目标是 SINT 或 INT 标签，则控制器通常会将该值转换回 SINT 或 INT 值
与 SINT 或 INT 的转换自动进行，无需额外编程。不过将占用额外的执行时间和内存</td></tr>
<tr><td>5）限制标签名称为 40 个字符</td><td colspan="2">下面是标签名称的规则：
仅字母字符（A-Z 或 a-z）、数字字符（0-9）和下划线（_）
必须以字母字符或下划线开始　不能多于 40 个字符
不能连续出现下划线字符（_）或以下划线字符（_）结尾
不区分大小写</td></tr>
<tr><td rowspan="6">6）使用混合大小写</td><td colspan="2">尽管标签不区分大小写（大写 A 和小写 a 相同），但混合大小写更容易阅读</td></tr>
<tr><td>这些标签更容易阅读：</td><td>相比这些标签：</td></tr>
<tr><td>Tank_1</td><td>TANK_1</td></tr>
<tr><td>Tank1</td><td>TANK1</td></tr>
<tr><td></td><td>tank_1</td></tr>
<tr><td></td><td>tank1</td></tr>
<tr><td rowspan="5">7）考虑标签的字母顺序</td><td colspan="2">RSLogix 5000 软件以字母顺序显示相同范围的标签。要更容易监视相关标签，请对要保存在一起的标签使用类似的开始字符</td></tr>
<tr><td>对罐的每个标签以 Tank 开始将标签保存在一起</td><td>否则，标签可能彼此分隔</td></tr>
<tr><td>标签名称</td><td>标签名称</td></tr>
<tr><td>Tank_North</td><td>North_Tank</td></tr>
<tr><td>Tank_South</td><td>South_Tank</td></tr>
</table>

2. 在标签编辑器中单击 N7 旁边的 +，如图 9-3 所示。

这将展开 N7 数组，显示其中的所有元素。可注意到从 N7［0］到 N7［7］的数据已定义。

然后是从 N7［20］到 N7［27］和从 N7［40］到 N7［47］。这些区域分别对应 3 个传送带，是传统的 PLC 数据表布局。

Scope: Conveyor_Progra　Show...　Show All

Name	Value	Force Mask	Style	Data Type	Description
N7	{...}	{...}	Decimal	INT[64]	
N7[0]	0		Decimal	INT	Motor Speed Low
N7[1]	0		Decimal	INT	Motor Speed High
N7[2]	0		Decimal	INT	Motor Speed Scaled Low
N7[3]	0		Decimal	INT	Motor Speed Scaled High
N7[4]	0		Decimal	INT	Motor Speed Scaled
N7[5]	0		Decimal	INT	Motor Speed Feedback Low
N7[6]	0		Decimal	INT	Motor Speed Feedback High
N7[7]	0		Decimal	INT	Motor Feedback
N7[8]	0		Decimal	INT	
N7[9]	0		Decimal	INT	
N7[10]	0		Decimal	INT	
N7[11]	0		Decimal	INT	
N7[12]	0		Decimal	INT	
N7[13]	0		Decimal	INT	
N7[14]	0		Decimal	INT	
N7[15]	0		Decimal	INT	
N7[16]	0		Decimal	INT	
N7[17]	0		Decimal	INT	
N7[18]	0		Decimal	INT	
N7[19]	0		Decimal	INT	
N7[20]	0		Decimal	INT	Motor Speed Low
N7[21]	0		Decimal	INT	Motor Speed High
N7[22]	0		Decimal	INT	Motor Speed Scaled Low

图 9-3　展开后的 N 数组

3. 仔细查看 B3、F8 和 T4。

它们都将采用相同的格式：传送带 1 从［0］开始，传送带 2 从［20］开始，传送带 3 从［40］开始。这样做的目的是在标签数据库中查找传送带数据更加容易。但是为传送带使用了 4 个数据表，浪费了内存。

通过创建 UDT 可更好地实现上述需求。现在就来创建 UDT。

4. 在控制器项目管理器中的“数据类型”（Data Types）文件夹下，单击“用户自定义”（User-Defined）旁边的 +，如图 9-4 所示。

将注意到这里有一个名为 Conveyor 的 UDT（已为启动 UDT 的创建），如图 9-5 所示。

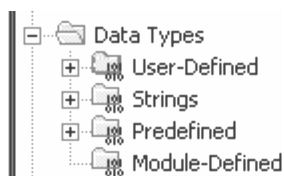


图 9-4　数据类型列表

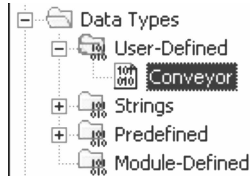


图 9-5　选择“Conveyor”用户自定义数据

5. 双击 Conveyor，如图 9-6 所示。

这将为 Conveyor 数据类型打开 UDT 编辑器。可以看到，UDT 混合不同的数据类型。在编程时这是非常有用的工具。

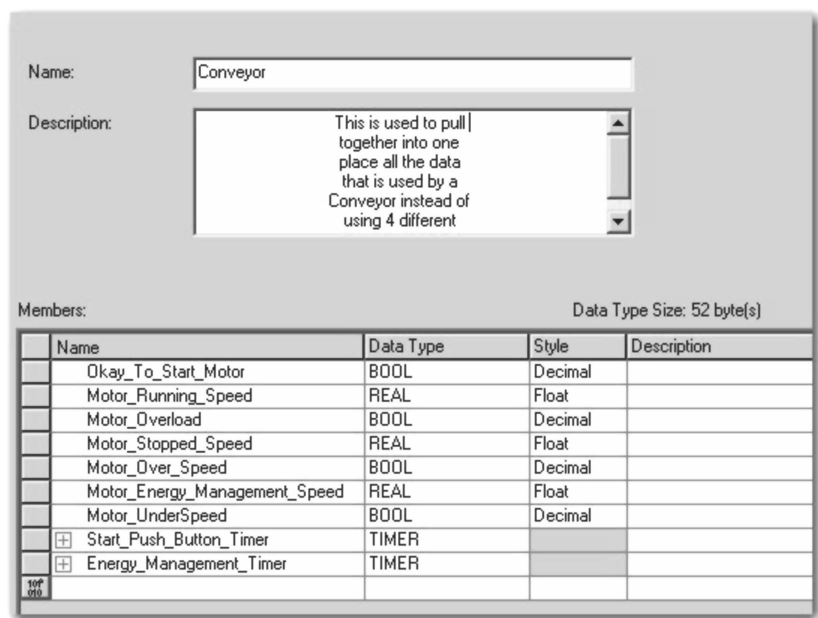


图 9-6　打开的 UDT 编辑器界面

参考信息

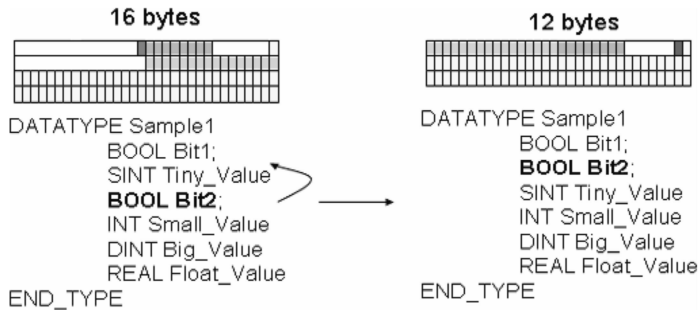
用户自定义数据类型的指导原则

创建用户自定义数据类型时，请记住以下指导原则：

如果包括表示 I/O 设备的数据，则必须使用逻辑将数据复制到 UDT 中。UDT 中不可为 I/O 指定别名。

如果要在 UDT 中包括数组，应将数组限制为一维。UDT 不支持多维数组。

创建 UDT 时，将类似的数据类型组合到一起。这将创建更为紧凑的 UDT。



查看 UDT 后，会注意到类似的数据类型并未组合到一起。BOOL 和 REAL 数据类型混在一起。UDT 的当前大小是 52 个字节。将 BOOL 组合在一起可降低这一数字。

6. 要进行此操作，单击 Motor_Overload 的左侧，如图 9-7 所示。这样将选中整行。

7. 单击“向上移动”（Move Up）按钮，如图 9-8 所示。这会在 UDT 中将 Motor_Overload 标签向上移动。

8. 对其他 BOOL 标签重复此操作，直到它们均组合到一起。

9. 单击“应用”（Apply）。请注意 UDT 的大小。已节省了 12 个字节。

Members: Data Type Size: 52 byte(s)

	Name	Data Type	Style	Description
	Okay_To_Start_Motor	BOOL	Decimal	
	Motor_Running_Speed	REAL	Float	
	Motor_Overload	BOOL	Decimal	
	Motor_Stopped_Speed	REAL	Float	
	Motor_Over_Speed	BOOL	Decimal	
	Motor_Energy_Management_Sp	REAL	Float	
	Motor_UnderSpeed	BOOL	Decimal	
	⊞ Start_Push_Button_Timer	TIMER		
	⊞ Energy_Management_Timer	TIMER		
*				

图 9-7 选中“Motor_Overload”标签

Members: Data Type Size: 52 byte(s)

	Name	Data Type	Style	Description
	Okay_To_Start_Motor	BOOL	Decimal	
	Motor_Overload	BOOL	Decimal	
	Motor_Running_Speed	REAL	Float	
	Motor_Stopped_Speed	REAL	Float	
	Motor_Over_Speed	BOOL	Decimal	
	Motor_Energy_Management_Sp	REAL	Float	
	Motor_UnderSpeed	BOOL	Decimal	
	⊞ Start_Push_Button_Timer	TIMER		
	⊞ Energy_Management_Timer	TIMER		
*				

图 9-8 向上移动后的“Motor_Overload”标签

10. 完成对 BOOL 标签的移动后，验证 UDT 的显示如图 9-9 所示。

Name: Conveyor

Description: This is used to pull together into one place all the data that is used by a Conveyor instead of using 4 different

Members: Data Type Size: 40 byte(s)

	Name	Data Type	Style	Description
	Okay_To_Start_Motor	BOOL	Decimal	
	Motor_Overload	BOOL	Decimal	
	Motor_Over_Speed	BOOL	Decimal	
	Motor_UnderSpeed	BOOL	Decimal	
	Motor_Running_Speed	REAL	Float	
	Motor_Stopped_Speed	REAL	Float	
	Motor_Energy_Management_Speed	REAL	Float	
	⊞ Start_Push_Button_Timer	TIMER		
	⊞ Energy_Management_Timer	TIMER		

图 9-9 完成 BOOL 标签的移动后数据大小

11. 回到“控制器标签”（Controller Tags），展开 N7 标签并注意 N7［0］到 N7［7］及其描述。
12. 使用标签的描述将这些标签添加到 UDT。手动执行行此操作（并要记住参考前文中的指

导原则 4）。

13. 完成后，单击“应用”（Apply）接受所做的更改，并验证 UDT 应类似图 9-10 所示。

Members:　　Data Type Size: 72 byte(s)

Name	Data Type	Style	Description
Okay_To_Start_Motor	BOOL	Decimal	
Motor_Overload	BOOL	Decimal	
Motor_Over_Speed	BOOL	Decimal	
Motor_UnderSpeed	BOOL	Decimal	
Motor_Running_Speed	REAL	Float	
Motor_Stopped_Speed	REAL	Float	
Motor_Energy_Management_Sp	REAL	Float	
⊞ Start_Push_Button_Timer	TIMER		
⊞ Energy_Management_Timer	TIMER		
Motor_Speed_Low	DINT	Decimal	
Motor_Speed_High	DINT	Decimal	
Motor_Scaled_Low	DINT	Decimal	
Motor_Scaled_High	DINT	Decimal	
Motor_Speed_Scaled	DINT	Decimal	
Motor_Speed_Fdbk_Low	DINT	Decimal	
Motor_Speed_Fdbk_High	DINT	Decimal	
Motor_Feedback	DINT	Decimal	
*			

图 9-10　程序中定义传送的完整 UDT

现在已拥有一个在程序中定义传送带的完整 UDT。

14. 单击“确定”（OK），关闭 UDT 编辑窗口。

15. 在“文件”（File）菜单中，选择“关闭”（Close）。软件将询问是否要保存对文件所做的更改，选择“否”（No）。

四、使用传送带 UDT

现在将使用刚创建的 UDT 传送带。意识到 UDT 将帮助优化标签数据库，并使代码的故障处理和开发更容易。

1. 在 RSLogix 5000 的工具栏菜单中，选择“文件 > 打开”（File > Open）。

2. 双击名为 Conveyor_Program_S4. ACD 的现有项目文件。这样将在 RSLogix 5000 中启动该项目。

3. 在控制器项目管理器中，双击“控制器标签”（Controller Tags），如图 9-11 所示。

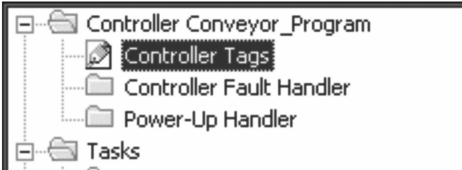


图 9-11　选中“控制器”标签

这将打开标签编辑器。查看标签编辑器，会发现数据库中有一个新标签 Conveyor_1，如图 9-12 所示。

Scope: Conveyor_Progra　Show...　Show All

Name	Value	Force Mask	Style	Data Type	Description
⊞-B3	{...}	{...}	Decimal	BOOL[64]	
⊞-Conveyor_1	{...}	{...}		Conveyor	This is used pull to...
⊞-F8	{...}	{...}	Float	REAL[64]	
⊞-F9	{...}	{...}	Float	REAL[64]	
⊞-N7	{...}	{...}	Decimal	INT[64]	
⊞-PD	{...}	{...}		PID	
⊞-T10	{...}	{...}		TIMER[64]	
⊞-T4	{...}	{...}		TIMER[64]	

图 9-12　打开的标签编辑器界面

4. 单击 Conveyor_1 旁边的 + 展开该标签，如图 9-13 所示。可注意到标签的格式与刚创建的 UDT 相同。这可通过查看“类型”（Type）列进行确认。

Scope: Conveyor_Progra Show... Show All

Name	Value	Force Mask	Style	Data Type
+B3	{...}	{...}	Decimal	BOOL[64]
−Conveyor_1	{...}	{...}		Conveyor
Conveyor_1.Okay_To_Start_Motor	0		Decimal	BOOL
Conveyor_1.Motor_Overload	0		Decimal	BOOL
Conveyor_1.Motor_Over_Speed	0		Decimal	BOOL
Conveyor_1.Motor_UnderSpeed	0		Decimal	BOOL
Conveyor_1.Motor_Running_Speed	0.0		Float	REAL
Conveyor_1.Motor_Stopped_Speed	0.0		Float	REAL
Conveyor_1.Motor_Energy_Management_Speed	0.0		Float	REAL
+Conveyor_1.Start_Push_Button_Timer	{...}	{...}		TIMER
+Conveyor_1.Energy_Management_Timer	{...}	{...}		TIMER
+Conveyor_1.Motor_Speed_Low	0		Decimal	DINT
+Conveyor_1.Motor_Speed_High	0		Decimal	DINT
+Conveyor_1.Motor_Scaled_Low	0		Decimal	DINT
+Conveyor_1.Motor_Scaled_High	0		Decimal	DINT
+Conveyor_1.Motor_Speed_Scaled	0		Decimal	DINT
+Conveyor_1.Motor_Speed_Fdbk_Low	0		Decimal	DINT
+Conveyor_1.Motor_Speed_Fdbk_High	0		Decimal	DINT
+Conveyor_1.Motor_Feedback	0		Decimal	DINT

图 9-13 展开的“Conveyor_1”列表

5. 在控制器项目管理器中，单击各个程序的 +，如图 9-14 展开连续任务中的 3 个传送带程序。

6. 双击例程 C2_Start_Stop，如图 9-15 所示。这将打开该例程，并可进行查看和编辑操作。

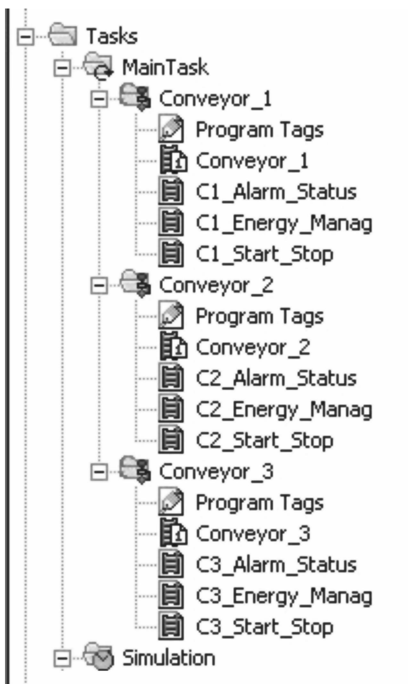


图 9-14 展开的 3 个传送带程序



图 9-15 选中“C2_Start_Stop”标签

7. 现在显示的是梯形图例程 C2_Start_Stop，如图 9-16 所示。可注意到，在此实例中 Conveyor_2 的寻址方式仍使用的是传统 PLC 的寻址方式。例如，在梯级 0 上，定时器的地址为 T4［20］。

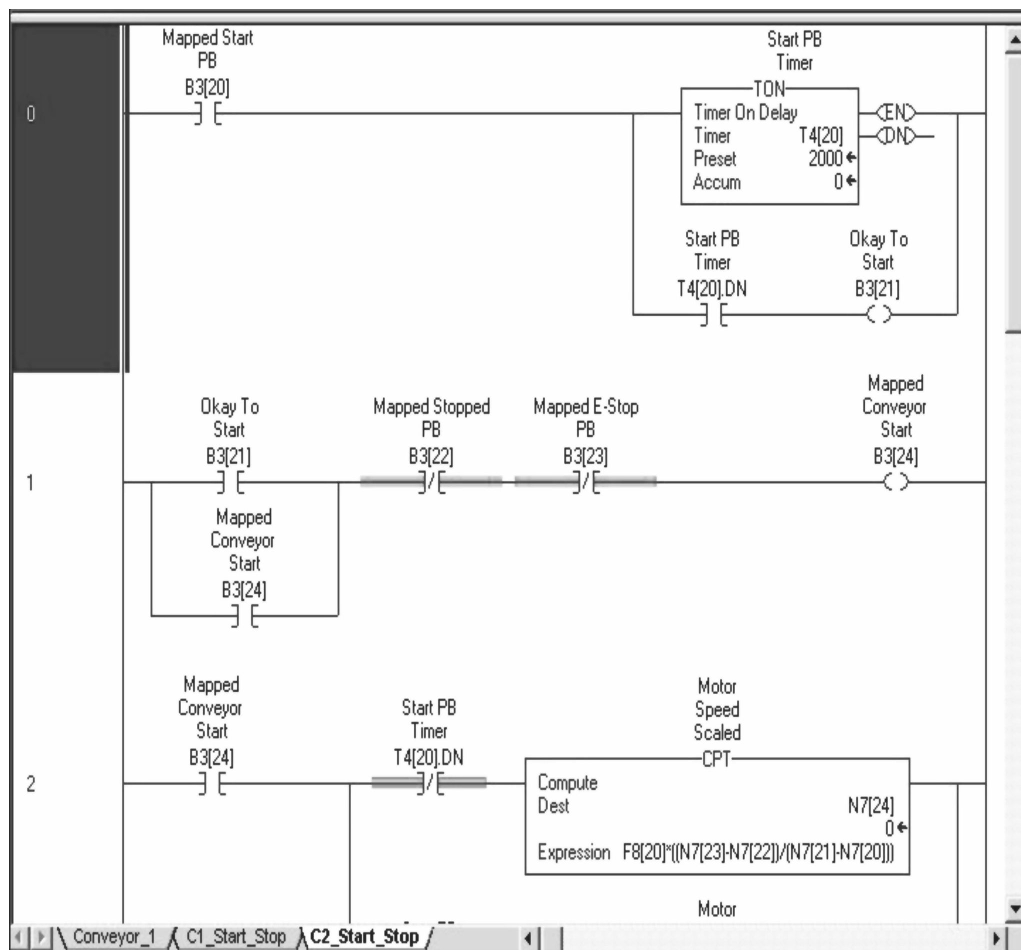


图 9-16 打开的例程梯形图

8. 双击例程 C1_Start_Stop，如图 9-17 所示。这将打开该例程，并可进行查看和编辑操作。

9. 现在显示的是梯形图文件 C1_Start_Stop，如图 9-18 所示。可看到传送带 1 的寻址方式使用的是 UDT。如图 9-19 所示，代码现在可以自文档化，而且标签名称可描述其具体的意义。无需再像传统 PLC 中一样添加地址描述。代码中仍有一些较早样式的地址，将稍后处理它们。

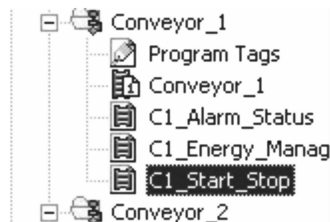


图 9-17 选中“C1_Start_Stop”标签

现在，继续更新其他两个传送带程序以使用 UDT。但首先需要创建两个新标签。

10. 在工具栏菜单中，选择“窗口 > 控制器标签-Conveyor_Program（控制器）”（Window > Controller Tags-Conveyor_Program（controller）），如图 9-19 所示。这会将标签编辑器窗口置于顶层。

11. 在标签编辑器窗口的左下角处，单击“编辑标签”（Edit Tags）选项卡；如图 9-20 所示。

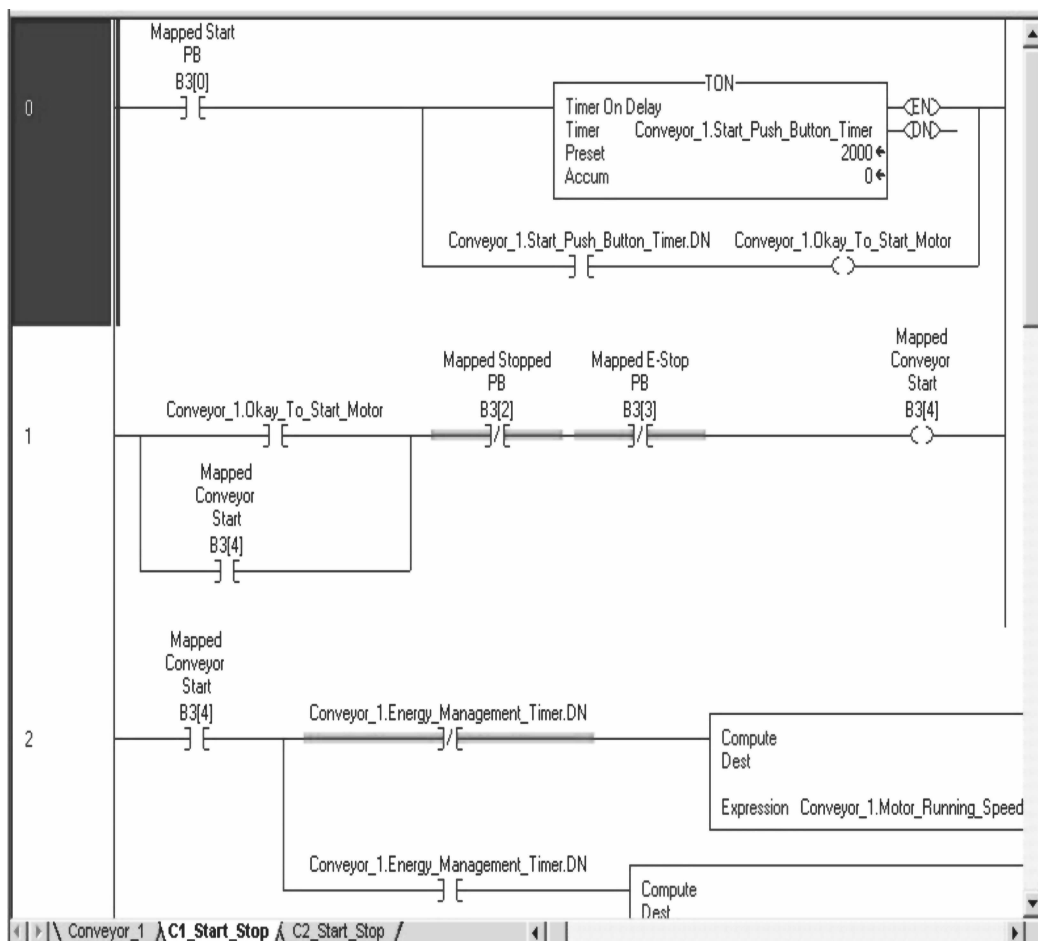


图 9-18 更新后的例程梯形图

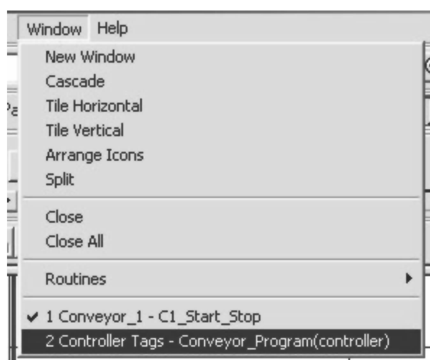


图 9-19 选中“Conveyor_Program（controller）”置于窗口顶层

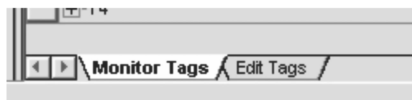


图 9-20 打开的标签编辑器

在这里将创建两个新标签，Conveyor_2 和 Conveyor_3。

12. 在标签编辑器中列表的底部，可注意到那里有一个开放空间，可创建一个新标签。

13. 在“名称:”（Name:）列中，输入“Conveyor_ 2”并在“类型”（Type）列中输入“Conveyor”，如图 9-21 所示。这样便会创建一个传送带类型的名称为 Conveyor_2 的标签。

⊞-T4			TIMER[64]	
Conveyor_2			Conveyor	Decimal

只要记录中仍可看到铅笔图标，便仍处于编辑模式下

图 9-21　创建的新标签“Conveyor_2”

14. 要创建新标签，单击 Conveyor_2 下面的新空白行，并验证标签编辑器的显示如图 9-22 所示。

⊞-T4			TIMER[64]	
⊞-Conveyor_2			Conveyor	

图 9-22　选中空白行

15. 使用与上述相同的方法，为 Conveyor_3 创建标签。
16. 完成后，验证标签数据库的显示如图 9-23 所示。

Name	Alias For	Base Tag	Data Type	Style
⊞-B3			BOOL[64]	Decimal
⊞-Conveyor_1			Conveyor	
⊞-F8			REAL[64]	Float
⊞-F9			REAL[64]	Float
⊞-N7			INT[64]	Decimal
⊞-PD			PID	
⊞-T10			TIMER[64]	
⊞-T4			TIMER[64]	
⊞-Conveyor_2			Conveyor	
⊞-Conveyor_3			Conveyor	

图 9-23　新的标签数据库

现在将更新梯形图代码并使用这些新标签。将从 Conveyor_2 开始。

仅修改 Alarm_Status 和 Energy_Manag 例程，如图 9-24 所示。使用表 9-2 并参考 Conveyor_1 例程。

Members:　　Data Type Size: 72 byte(s)

Name	Data Type	Style	Description
Okay_To_Start_Motor	BOOL	Decimal	
Motor_Overload	BOOL	Decimal	
Motor_Over_Speed	BOOL	Decimal	
Motor_UnderSpeed	BOOL	Decimal	
Motor_Running_Speed	REAL	Float	
Motor_Stopped_Speed	REAL	Float	
Motor_Energy_Management_Sp	REAL	Float	
⊞ Start_Push_Button_Timer	TIMER		
⊞ Energy_Management_Timer	TIMER		
Motor_Speed_Low	DINT	Decimal	
Motor_Speed_High	DINT	Decimal	
Motor_Scaled_Low	DINT	Decimal	
Motor_Scaled_High	DINT	Decimal	
Motor_Speed_Scaled	DINT	Decimal	
Motor_Speed_Fdbk_Low	DINT	Decimal	
Motor_Speed_Fdbk_High	DINT	Decimal	
Motor_Feedback	DINT	Decimal	
*			

图 9-24　更新后的梯形图代码标签

表　9-2

N7[20]	0		Decimal	INT	Motor Speed Low
N7[21]	0		Decimal	INT	Motor Speed High
N7[22]	0		Decimal	INT	Motor Speed Scaled Low
N7[23]	0		Decimal	INT	Motor Speed Scaled High
N7[24]	0		Decimal	INT	Motor Speed Scaled
N7[25]	0		Decimal	INT	Motor Speed Feedback Low
N7[26]	0		Decimal	INT	Motor Speed Feedback High
N7[27]	0		Decimal	INT	Motor Feedback
T4	{...}	{...}		TIMER[64]	
T4[0]	{...}	{...}		TIMER	Start PB Timer
T4[1]	{...}	{...}		TIMER	Energy Manag Timer
T4[2]	{...}	{...}		TIMER	
T4[3]	{...}	{...}		TIMER	
B3[20]	0		Decimal	BOOL	Mapped Start PB
B3[21]	0		Decimal	BOOL	Okay To Start
B3[22]	0		Decimal	BOOL	Mapped Stopped PB
B3[23]	0		Decimal	BOOL	Mapped E-Stop PB
B3[24]	0		Decimal	BOOL	Mapped Conveyor Start
B3[25]	0		Decimal	BOOL	Mapped Entrance PE
B3[26]	0		Decimal	BOOL	Mapped Exit PE
B3[27]	0		Decimal	BOOL	Mapped Motor Overload
B3[28]	0		Decimal	BOOL	C2 Motor Overload
B3[29]	0		Decimal	BOOL	C2 Motor Over Speed
B3[30]	0		Decimal	BOOL	C2 Motor Under Speed

让从一个示例开始。

17. 在控制器项目管理器中，双击例程 C2_Alarm_Status。

重要说明：

目前，将忽略任何在描述中包括文字“映射”的 B3 地址。

18. 在梯级零上，注意描述为 C2 电机过载的 OTE 指令，如图 9-25 所示。通过查看前一页中的图，可发现此描述与 UDT 中的第二个成员相匹配。



图 9-25　匹配后的 C2 电机过载 OTE 指令

19. 要更改地址，双击 B3［28］，如图 9-26 所示。这将打开一个下拉框，以便选择一个新标签。

20. 单击向下箭头（这将打开标签浏览器），滚动浏览列表直到找到 Conveyor_2。

21. 单击 + 展开该标签，然后选择 Conveyor_2. Motor_Overload，如图 9-27 所示。

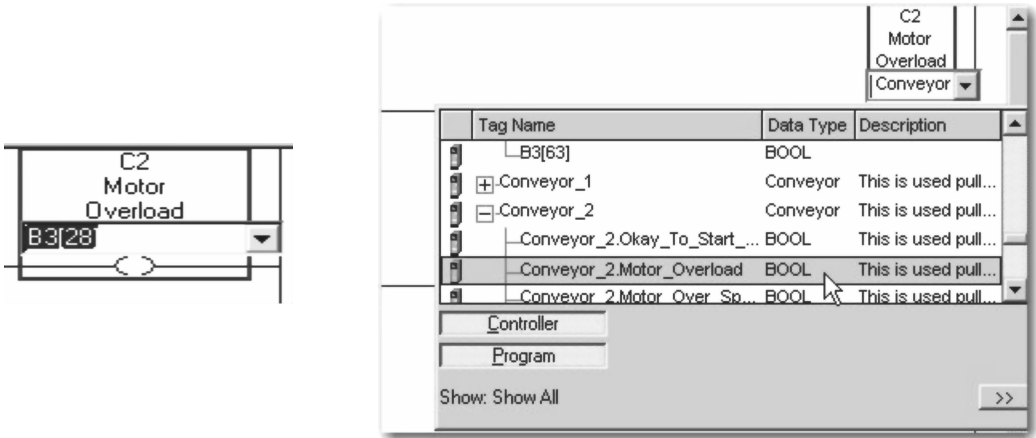


图 9-26 选中 B3［28］ 图 9-27 展开的标签列表

22. 双击 Conveyor_2. Motor_Overload。将会使 Conveyor_2. Motor_Overload 成为 OTE 指令的新标签。

23. 单击标签编辑器的空白处，验证梯级的显示，如图 9-28 所示，（请再次注意自文档化的代码）。



图 9-28 更新后的标签描述信息

24. 现在，继续并完成此例程的更新。然后更新传送带 2 的 Energy_Manag 例程。

25. 为传送带 3 重复相同的步骤。

26. 可看到到标签 Conveyor_2. Motor_Overload 确实有描述。对于版本 V12 和较早版本的 RSLogix 5000，在 UDT 编辑器中输入的描述并不会在 UDT 编辑器以外显示。

参考信息

UDT 传递和附加描述选项

从 RSLogix 5000 的版本 13 开始，“工作站选项”（Workstation Options）下提供两个新设置，如图 9-29 所示。

如果启用传递描述，则只要不为标签提供显式描述：

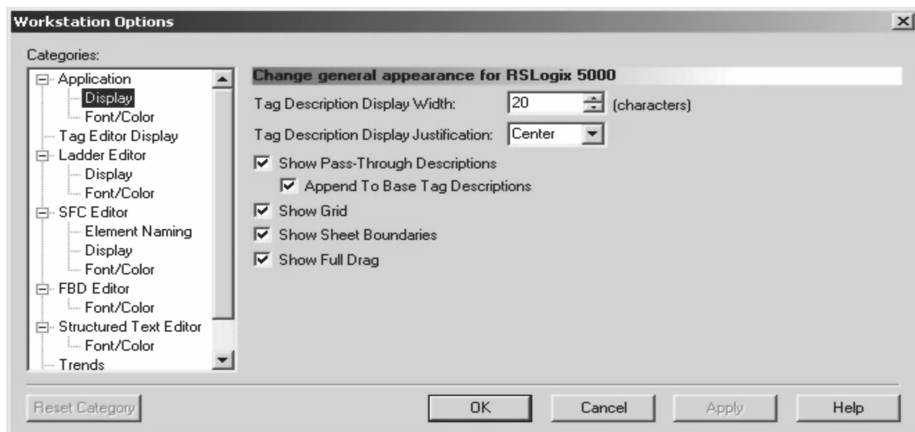


图 9-29　打开的“工作站选项”界面

1）在标签是别名时将会显示基础标签的描述，或者；

2）在标签是成员时将会显示定义的数据类型的成员描述，或者；

3）在标签是位、数组元素或成员时将会显示基础标签的描述，或者；

4）将会显示定义的基本数据类型的描述。

如果选中此复选框，“附加到基础标签描述”（Append to Base Tag Descriptions）复选框便会启用。如果要使传递描述中包含基础标签描述，则选中“附加到基础标签描述”（Append to Base Tag Descriptions）复选框。根据传递描述规则添加任意附加描述。在提供此功能之前，用户必须手动从 UDT 编辑器中剪切描述，并在标签编辑器里手动将这些描述粘贴到标签中。这样会使编程工作多花费许多个小时。

来了解此新功能的工作方式。

27. 在控制器项目管理器中的“数据类型”（Data Types）文件夹下，单击“用户自定义”（User-Defined）旁边的 +，如图 9-30 所示。

将注意到有一个名为 Conveyor 的 UDT，如图 9-31 所示。

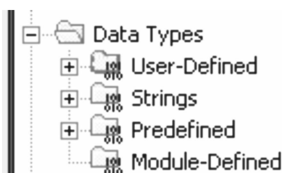


图 9-30　项目管理器列表

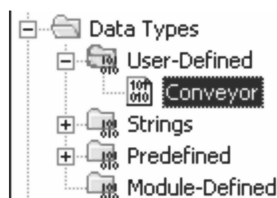


图 9-31　选择“Conveyor”

28. 双击文字 Conveyor，如图 9-32 所示。这将为 Conveyor 数据类型打开 UDT 编辑器。

29. 为 UDT 的成员添加一些描述，完成后单击“应用”（Apply）。

30. 更改 UDT 的描述，如图 9-33 所示。

31. 单击“应用”（Apply），然后单击“确定”（OK），关闭 UDT 编辑器。

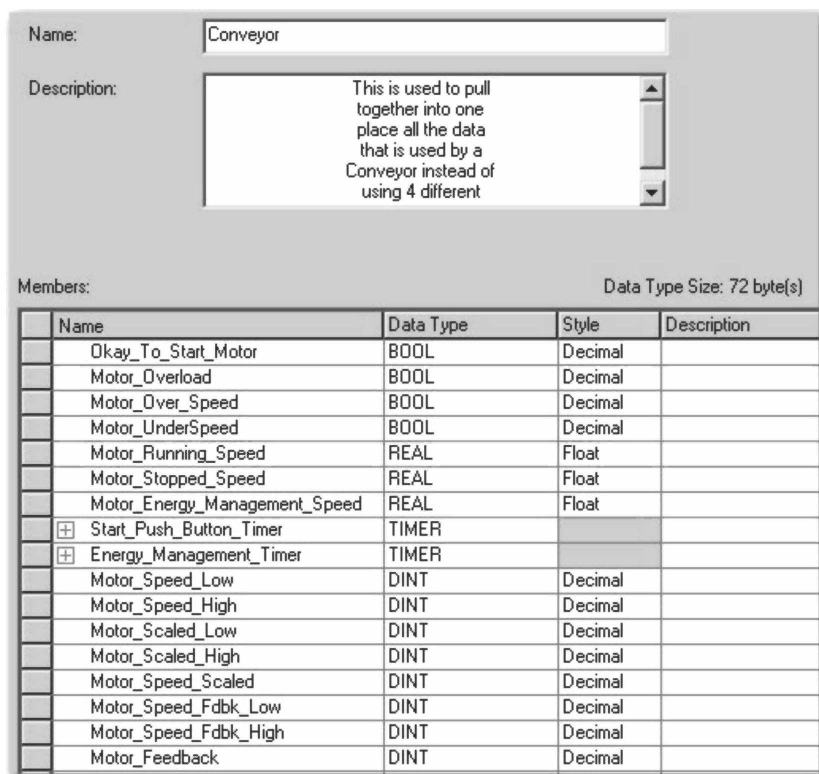


图 9-32 打开的 UDT 编辑器界面

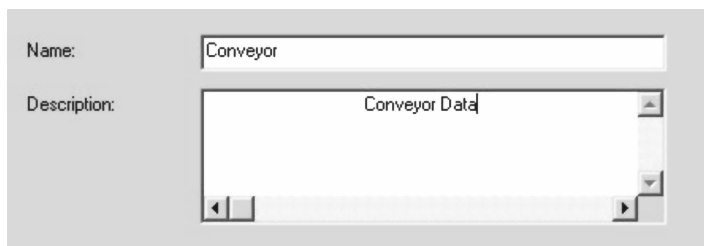


图 9-33 更改 UDT 描述信息

32. 在控制器项目管理器中，双击例程 C2_Alarm_Status，如图 9-34 所示。

现在可看到，输入的描述现在已显示在标签上。该描述包括 UDT 描述（传送带数据）和 UDT 成员描述（电动机过载指示）。

33. 在工具栏菜单中，选择“工具 > 选项”（Tools > Options），如图 9-35 所示。

将出现“选项”（Options）窗口，如图 9-36 所示。

34. 在“应用程序”（Application）下，单击“显示”（Display）。窗口如图 9-37 所示。

35. 调整“工作站选项”（WorkStation Options）窗口的位置，以便既可看见该窗口同时又可看到 C2_Alarm_Status 例程中的 OTE 指令。

36. 单击“附加到基础标签描述”（Append To Base Tag Descriptions）旁边的复选框关闭此选项，然后单击“应用”（Apply）接受所做的更改，如图 9-38 所示。

可以看到到描述已更改。UDT 描述“传送带数据”已移除。

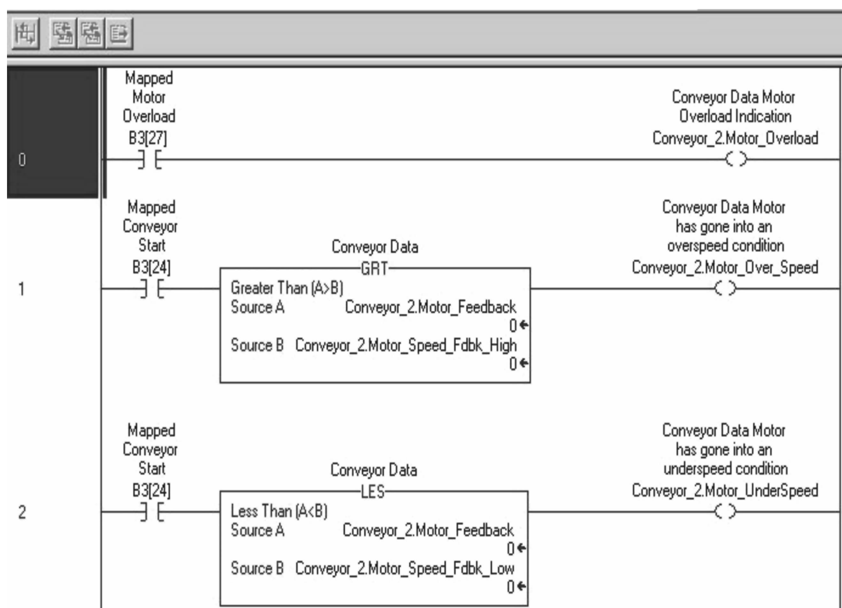


图 9-34 标签上显示的描述信息

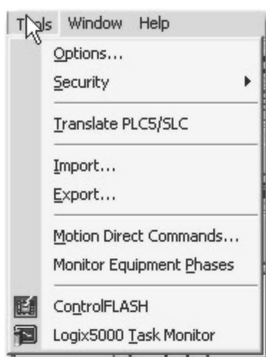


图 9-35 工具选项菜单

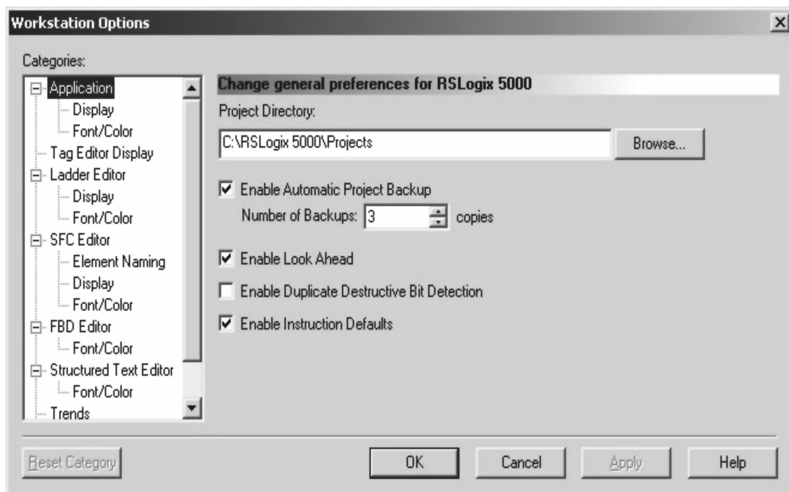


图 9-36 打开的“选项”（Options）窗口

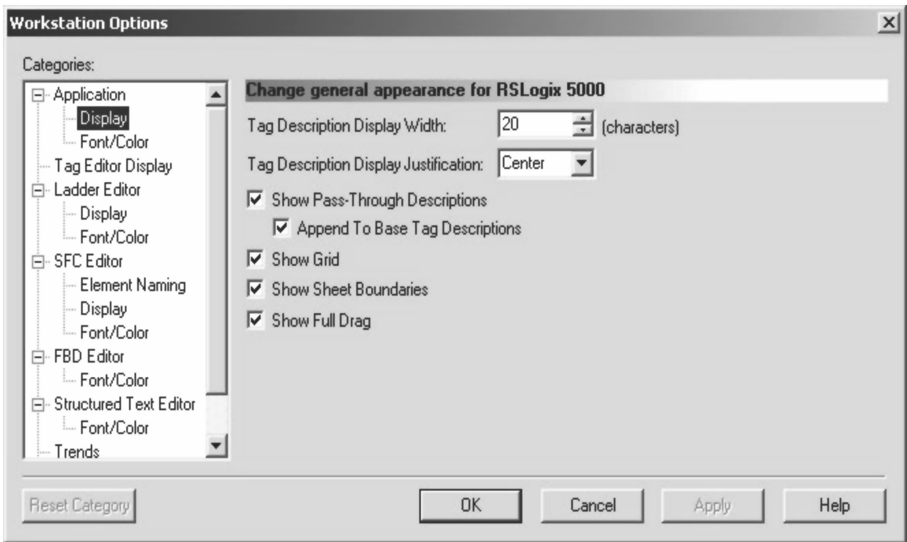


图 9-37 打开的“显示（Display）”选项界面

37. 单击“显示传递描述”（Show Pass-Through Descriptions）旁边的复选框，关闭此选项，然后单击“应用”（Apply）接受所做的更改，如图 9-39 所示。

可以看到描述已再次更改。描述处已没有任何内容。这是在版本 V12 和更早 RSLogix 5000 版本中梯级的显示方式。

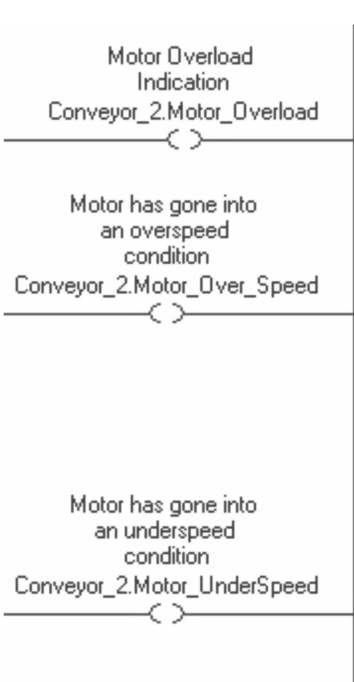


图 9-38 移除基础描述信息后的梯形图

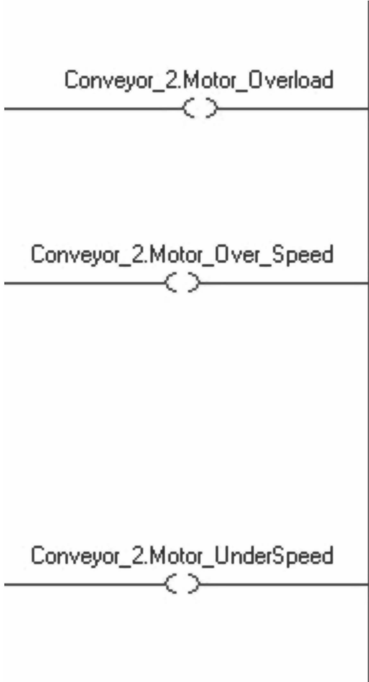


图 9-39 移除传递描述信息后的梯形图

38. 重新打开两个选项，接着单击“应用”（Apply），然后单击“确定”（OK）。

39. 完成后，单击“保存”（Save）图标。

40. 在“文件”（File）菜单中选择“关闭”（Close）。

五、使用数据范围划定

在本实验中，将了解数据范围划定的优势。

刚使用过的用户自定义数据类型，可以更好地组织标签数据库，并可创建自文档化的代码。接下来将要进行的最后一步是数据范围划定，将标签的范围划定为程序级别。通过数据范围划定如下：

1）对不同的程序可以使用相同的标签名称，因为各程序范围内的标签彼此隔离；

2）开发标准代码，可复制使用多次，无需担忧标签名称冲突；

3）不再需要管理标签名称和数据位置；

4）加速项目开发过程。

参考信息

数据范围划定

创建标签时，将其定义为控制器标签（全局数据）或特定程序的程序标签（本地数据）。

Logix 控制器允许将应用分割为多个程序。无需管理程序间标签名称的冲突问题，可以更轻松地在多个程序中重复使用代码和标签名称。

程序范围内的数据与其他程序隔离。例程不能访问其他程序的程序范围内的数据。可以在多个程序内重复使用程序范围标签的标签名称。

六、打开现有的控制器文件

在实验的该部分中，将打开一个现有程序。

1. 在计算机桌面上，双击 Lab Files 文件夹。

2. 双击名为 Conveyor_Program_S5. ACD 的现有项目。

将在 RSLogix 5000 中启动该项目。

七、对程序进行数据范围划定

1. 在控制器项目管理器中，双击“控制器标签”（Controller Tags），打开标签编辑器，如图 9-40 所示。

Scope: Conveyor_Progra   Show...   Show All

Name	Alias For	Base Tag	Data Type
+-B3			BOOL[64]
Control_Output			REAL
+-Conveyor_1			Conveyor
+-Conveyor_2			Conveyor
+-Conveyor_3			Conveyor
Process_Variable			REAL
+-Temp_Loop_PID			PID

图 9-40　打开的标签编辑器

2. 验证已处于“编辑标签”（Edit Tags）选项卡中（如果不是，则无法执行剪切/粘贴操作）。

打开标签编辑器后，将看到有许多旧标签不见了（N7、F8、T4）。在创建传送带 UDT 并修改代码后，便不再需要这些标签了。标签 Conveyor_1、Conveyor_2 和 Conveyor_3 仍存在。F9 和 PD 文件被更改为标签名称，而不是像传统 PLC 一样采用数据表编号的方式。在将 PID 移动到周期性任务中后，已将 T10 文件删除。

现在，要将标签 Conveyor_1、Conveyor_2 和 Conveyor_3 从控制器范围移动到程序范围。

3. 双击“数据类型”（Data Type）列，如图 9-41 所示。便会将标签排列为更适合于进行的编辑。

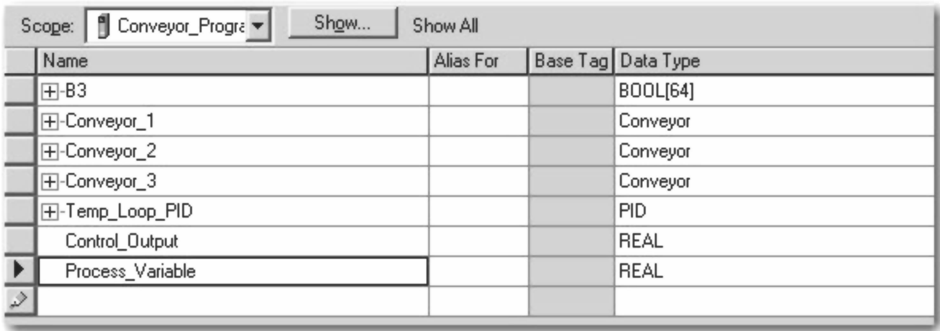


图 9-41　排列后的数据标签

首先，会将标签 Temp_Loop_PID、Control_Output 和 Process_Variable 移动到 Temperature_Loop 程序范围。

4. 单击标签 Temp_Loop_PID 的左侧，如图 9-42 所示。这样将选中整行。



图 9-42　选中目标标签

5. 按住 Shift 键并单击标签 Process_Variable 的左侧，如图 9-43 所示。这样将选中所有 3 个标签。

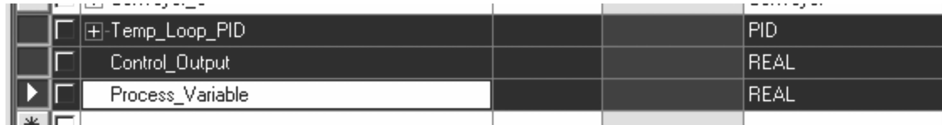


图 9-43　选中多行目标标签

6. 在工具栏菜单中，选择“编辑 > 剪切”（Edit > Cut），如图 9-44 所示。

7. 在“范围”（Scope）下拉菜单中，单击向下箭头。单击“程序”（Programs）旁边的 +，展开该标签，如图 9-45 所示。这将显示所有当前的控制器和程序标签数据库。

8. 选择 PID_Loop 范围，如图 9-46 所示。

这将会为程序 PID_Loop 的程序范围标签打开标签编辑器，如图 9-47 所示。

9. 单击打开的标签名称文件，以便光标出现在图 9-48 位置。

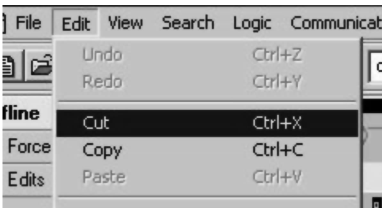


图 9-44　打开的标签编辑菜单

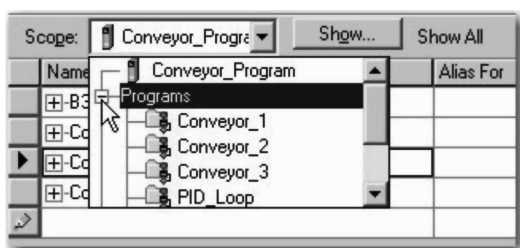


图 9-45　打开范围选择列表

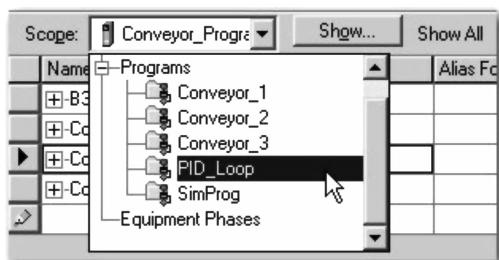


图 9-46　选择目标程序

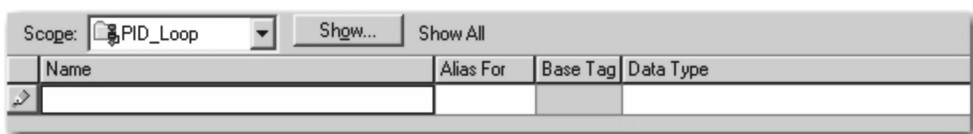


图 9-47　打开的目标程序标签编辑器

10. 在工具栏菜单中，选择“编辑 > 粘贴”（Edit > Paste），如图 9-49 所示。

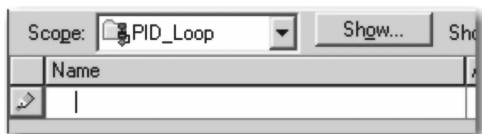


图 9-48　选中首行



图 9-49　选中标签编辑菜单 Paster

11. 验证 3 个标签已粘贴，如图 9-50 所示。

Name	Alias For	Base Tag	Data Type
⊞-Temp_Loop_PID			PID
Control_Output			REAL
Process_Variable			REAL

图 9-50　粘贴创建的新标签

12. 切换回“控制器范围”（Controller scope）标签。将显示如图 9-51 中界面。

Scope: Conveyor_Progra　Show...　Show All

Name	Alias For	Base Tag	Data Type
⊞-B3			BOOL[64]
⊞-Conveyor_1			Conveyor
⊞-Conveyor_2			Conveyor
⊞-Conveyor_3			Conveyor

图 9-51　打开的“控制器范围”标签

13. 将标签 Conveyor_1 剪切/粘贴到 Conveyor_1 程序范围中。

14. 将标签 Conveyor_2 剪切/粘贴到 Conveyor_2 程序范围中。

15. 将标签 Conveyor_3 剪切/粘贴到 Conveyor_3 程序范围中。

16. 完成后，验证 B3 是控制器范围中唯一所剩的标签，如图 9-52 所示。

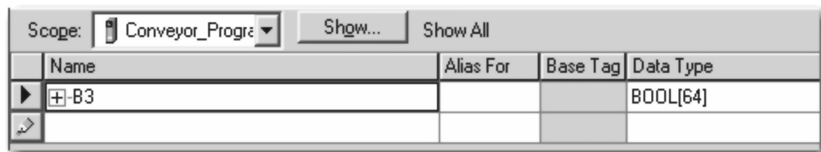


图 9-52　剪切完成后的控制器标签

17. 完成对标签的移动后，单击“校验控制器”（Verify Controller）图标。将会显示 4 个错误，但这属正常情况。这些错误与仿真器程序相关。

之前，提到过程序范围内的数据与其他程序隔离。例程不能访问其他程序的程序范围内的数据。但是，可以在多个程序内重复使用程序范围标签的标签名称。让其来证明这一点，进入所有 3 个传送带的程序范围标签，将标签名称 Conveyor_1、Conveyor_2 和 Conveyor_3 均更改为 Conveyor。

18. 在“范围”（Scope）下拉菜单中选择 Conveyor_1，如图 9-53 所示。

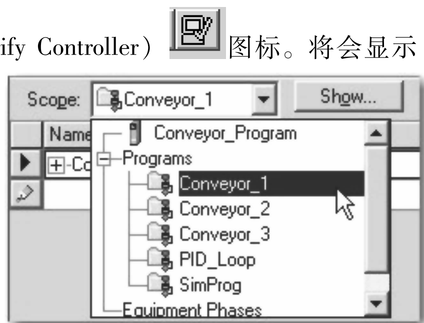


图 9-53　打开的“范围”（Scope）下拉菜单

19. 将标签名称“Conveyor_1”更改为“Conveyor”，如图 9-54 所示。

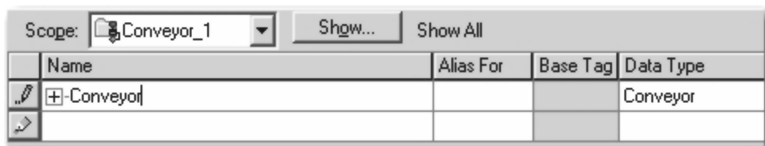


图 9-54　更名后的标签

20. 修改其他 2 个标签，将 Conveyor_2 和 Conveyor_3 也改为相同名称。

21. 标签名称更改完成后，单击“校验控制器”（Verify Controller）图标。

将会显示 4 个错误，但这属正常情况。这些错误与仿真器程序相关。重复的标签不会发生任何错误。这是因为各程序范围内的标签彼此隔离。

参考信息

对标签进行数据范围划定，这使得更容易在多个程序间重复使用代码和标签名称。在此情况下可创建使用程序范围标签的名为 Conveyor 的程序。然后，可根据传送带段数向项目中添加相应数量的此程序，无需再修改代码或标签。

22. 在工具栏菜单中，选择“文件 > 关闭”（File > Close）。

23. 当软件询问是否保存更改时，应选择“否”（No）。

实验 10：INT 与 DINT 数据类型

一、实验目的

1. 使用 INT 数据类型与 DINT 数据类型执行程序。
2. 使用 INT 数据类型与 DINT 数据类型确定程序大小。

二、打开现有控制器文件并将其下载到控制器中

1. 在计算机桌面上，双击 Lab Files 文件夹。
2. 双击 Z_A3_Int_Dint_CLX. ACD 文件，以在 RSLogix 5000 中启动该项目。
3. 在工具栏菜单中，选择“通信 > 在线节点”（Communication > Who Active）。
4. 浏览到 ControlLogix 演示箱插槽 1 中的控制器（请注意站上 1756-ENBT 模块的 IP 地址 10. 112. 2. 1x3，其中 x 是学员站的编号）。
5. 单击“下载”（Download）转到“远程运行”（Remote Run）模式。

三、使用 INT 数据类型与 DINT 数据类型执行

参考信息

INT 与 DINT 数据类型

所有 Logix 控制器都使用 32 位 RISC 处理器，该处理器已针对 32 位运算进行过优化。为此，非 32 位的数据类型需要额外地执行时间和内存开销。非 32 位数据类型为 BOOL、SINT 和 INT。因此，编程时请尽可能地使用 DINT。

1. 在控制器项目管理器中展开“主程序”（MainProgram），如图 10-1 所示。
2. 双击“主例程”（MainRoutine），如图 10-2 所示。这将打开例程，可供查看。

现在显示的是梯形图例程“主例程”（MainRoutine）。此例程可决定程序的执行方式。共有 2 个可调用的例程：Math_DINT 和 Math_INT。这两个例程完全相同，但一个采用的是 DINT 数据类型，另一个采用的是 INT 数据类型。当前正在执行的是 Math_INT 文件。

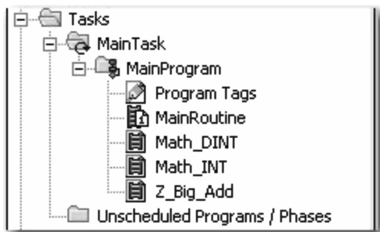


图 10-1 展开的项目管理器列表

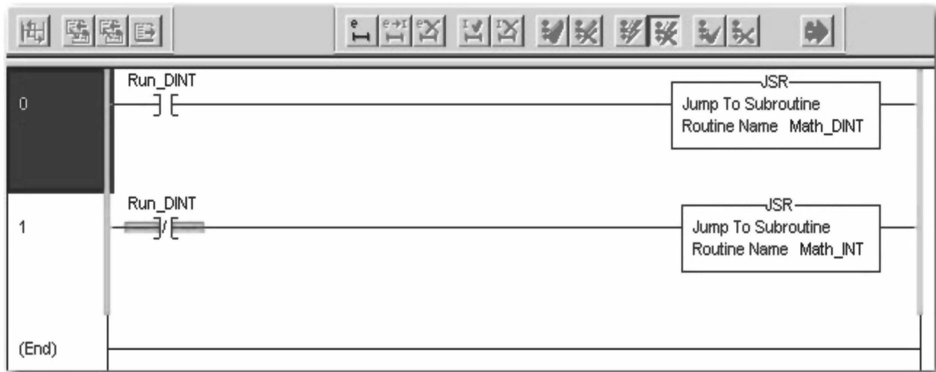


图 10-2 打开的主例程界面

3. 右键单击 MainTask 文件夹，并选择“属性”（Properties），如图 10-3 所示。

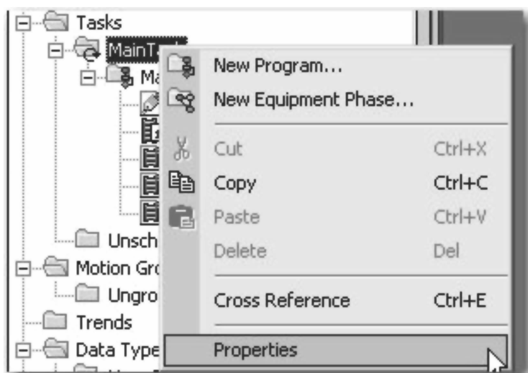


图 10-3　打开的项目编辑菜单

4. 出现“任务属性-主任务”（Task Properties-MainTask）窗口后，单击“监视”（Monitor）选项卡，如图 10-4 所示。

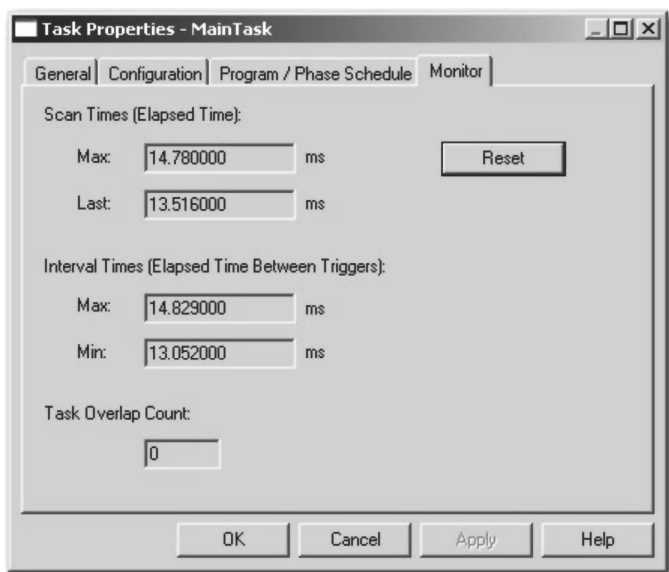


图 10-4　监视（Monitor）选项卡

5. 单击“重置”（Reset）按钮。

6. 当前，控制器正在执行 Math_INT 例程。扫描时间约为 14.5ms。请记住此例程使用的是 INT 数据类型。

7. 不要关闭“任务属性”（Task Properties）窗口。将“主例程”（MainRoutine）窗口置于顶层。

8. 右键单击梯级零上的标签 Run_DINT，并选择“切换位”（Toggle Bit），如图 10-5 所示。这将导致程序执行 Math_DINT，并停止执行 Math_INT。

9. 在工具栏菜单中，选择“窗口 > 任务属性 > 主任务”（Window > Task Properties-MainTask）。这会将“任务属性”（Task Properties）窗口置于顶层。

10. 单击“重置”（Reset）按钮。扫描时间现在约为 5ms，几乎快了 3 倍。

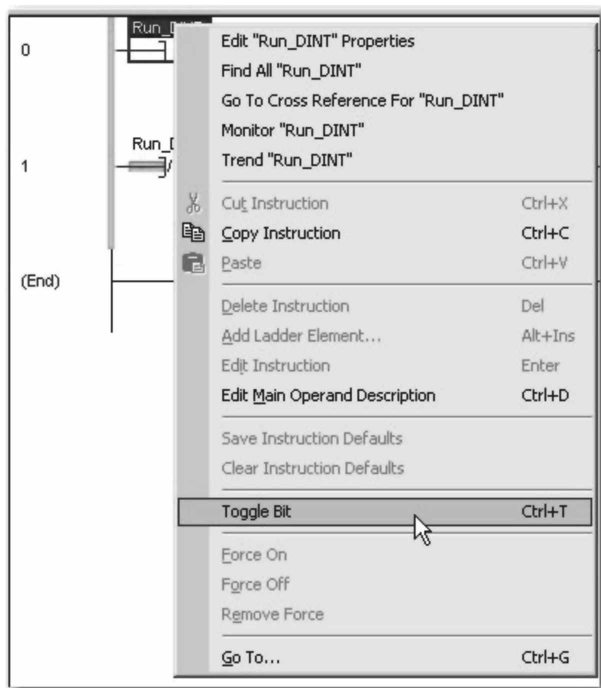


图 10-5 选择“切换位”选项

11. 单击“确定”（OK），关闭“任务属性”（Task Properties）窗口。

12. 在“通信”（Communication）菜单中选择“脱机”（Go Offline），如图 10-6 所示。

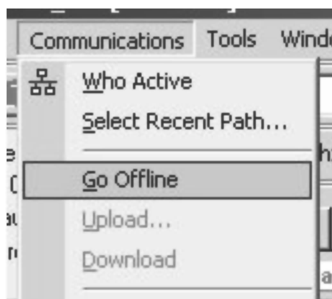


图 10-6 选择“脱机”（Go Offline）模式

四、理解使用 INT 数据类型与 DINT 数据类型，确定程序大小的效果

如前文所述，非 32 位的数据类型需要额外的执行时间和内存开销。由于，已了解了执行方面，现在来了解内存方面的情况。

1. 在控制器项目管理器中，将看到一个名为 Z_Big_Add 的例程，如图 10-7 所示。此例程有 30 个使用了 INT 数据类型的 Add 指令。

2. 在 RSLogix 5000 的控制器面板中，单击“控制器属性”（Controller Properties）按钮，如图 10-8 所示。

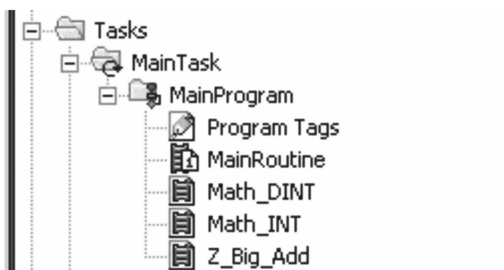


图 10-7 展开的项目管理器列表

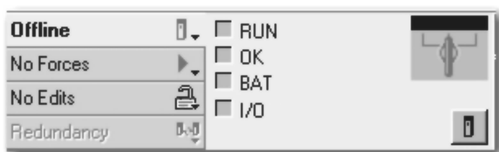


图 10-8 打开的控制器面板

3. 出现“控制器属性”（Controller Properties）窗口后，单击“内存”（Memory）选项卡，如图 10-9 所示。

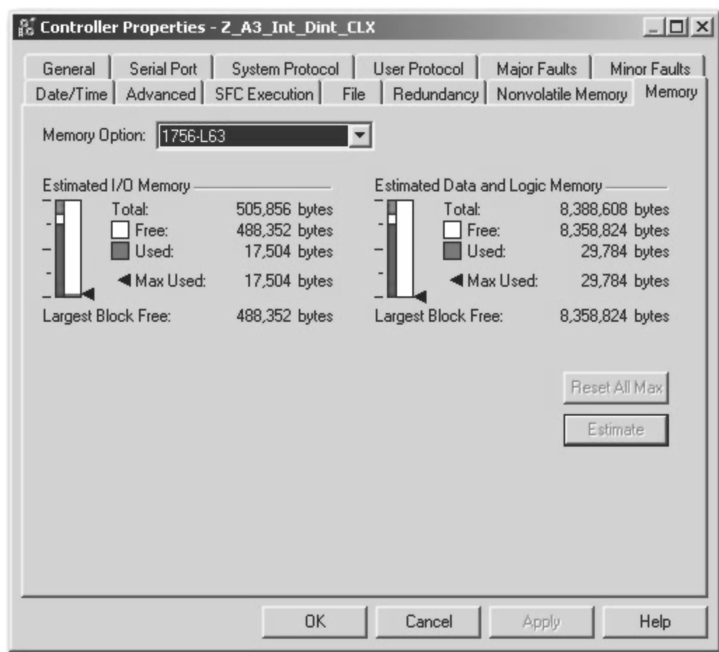


图 10-9　打开的“控制器属性”界面

4. 单击“评估”（Estimate）按钮。在“估计数据和逻辑内存”（Estimated Data and Logic Memory）下可看到已使用 38448 字节的内存。

5. 保持“控制器属性”（Controller Properties）窗口处于打开状态。

6. 在控制器项目管理器中，双击“控制器标签”（Controller Tags）打开标签编辑器。当标签编辑器出现时，确保在“编辑”（Edit）选项卡中。

7. 查看标签编辑器，将看到一个名为 Data_File_Int 的标签，如图 10-10 所示。此标签定义为整数数据类型。此标签当前用于 Z_Big_Add 例程中。

8. 将 Data_File_Int 的类型从 INT［1000］更改为 DINT［1000］。做出更改后，界面应类似于图 10-11 所示。

9. 关闭标签编辑器。将再次显示“控制器属性”（Controller Properties）窗口。

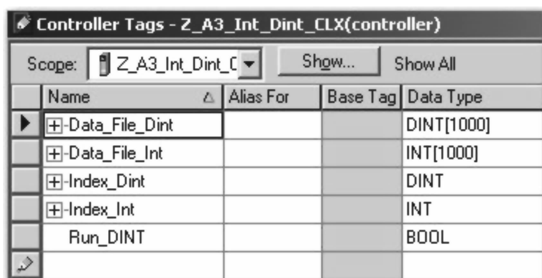


图 10-10　打开的例程标签编辑器

10. 再次单击“评估”（Estimate）按钮。现在在“估计数据和逻辑内存”（Estimated Data and Logic Memory）下可看到已使用 36724 字节的内存。比之前少了 1724 个字节。这是因为留出了额外的内存来保存临时转换的值。既然数据类型是 DINT，就无需再分配该部分内存。

11. 单击“确定”（OK），关闭“控制器属性”（Controller Properties）窗口。

12. 在工具栏菜单中，选择“文件 > 退出”（File > Exit），退出并且不保存所做的更改。

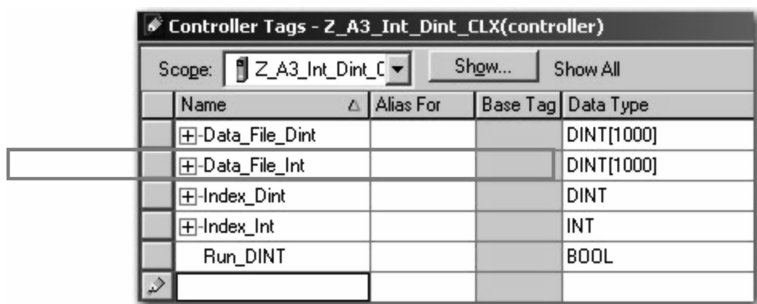


图 10-11　更改为 DINT 数据类型

五、实验小结

在本实验中，展示了非 32 位数据类型对程序大小的影响。程序编译后，系统便会分配附加内存。控制器需要通过内部寄存器来存储已转换类型的数据。在使用的示例程序中，共有 30 个采用了 INT 数据类型的 Add 指令，它们使程序大小增加 1724 个字节。

实验 11：使用任务和任务中的程序进行分段

一、实验目的

在这部分实验中，将了解以下概念：

在传统 PLC 中，使用任务来对项目进行分段与使用任务中的程序来对项目进行分段。有三种不同类型的“任务”：

1）主任务 = Logix 中的连续任务；

2）STI 任务 = Logix 中的周期性任务；

3）PII/DII 任务 = Logix 中的事件任务。

但是，只能有一个 STI 和 PII/DII。在 Logix 中可以有多个周期性任务和事件任务。这可以是一个功能强大的工具，但在执行前应先了解它。

在使用 Logix 编程时会遇到以下问题：对项目进行分段的最佳方法是什么？是在一个任务中使用多个程序更好？还是使用带有一个程序的多个周期性任务更好？最常见的回答是使用带有一个程序的多个周期性任务，这在 99% 的情况下都是错误的选择。为什么会这样？通过在线示例可以最好地回答这一问题。

将通过使用一个程序来说明，该程序有一个带有 5 个单独程序连续任务、5 个分别带有一个程序的周期性任务，以及一个带有 5 个程序的周期性任务。

对于连续任务，所有程序都相同，一个简单循环需要 5ms 的程序执行时间。因此，带有 5 个程序连续任务的执行时间约为 25ms。5 个周期性任务包含的程序与连续任务相同，因此执行时间也是每个 5ms。问题是现在变为周期性任务的运行速率应该是多少？任务的运行优先级是多少？

参考信息

周期性任务优先级

对于周期性任务，可指定从 1 到 15 的优先级。1 是最高的优先级，15 是最低的优先级。优先级较高的任务始终会中断优先级较低的任务。如果任务优先级相同时会如何呢？它们将进行任务切换。

任务切换的意思是，当优先级相同的多个任务执行时，每运行 1ms 时间片时任务之间便会进行切换。例如，如果有 3 个优先级相同的周期性任务，则任务 1 运行 1ms 后将切换到任务 2，任务 2 运行 1ms 后再切换到任务 3，任务 3 也将运行 1ms。然后一直按此模式进行下去，直到所有这些任务都执行完毕。

大部分人都会选择 21ms 或 25ms 的周期性任务周期，并会设置相同的优先级。

二、打开现有控制器文件并将其下载到控制器

在这部分实验中，将打开一个现有程序。

1. 在计算机桌面上，双击 Lab Files 文件夹。

2. 双击 Z_A1_program_task_CLX. ACD 文件以在 RSLogix 5000 中启动该项目，如图 11-1 所示。

3. 在工具栏菜单中，选择“通信 > 在线节点”（Communication > Who Active）。

4. 浏览到 ControlLogix 演示箱插槽 1 中的控制器（请注意站上 1756-ENBT 模块的 IP 地址 192. 168. 1. x + 10，其中 x 是学员站的编号）。

5. 单击“下载”（Download）转到“远程运行”（Remote Run）模式。

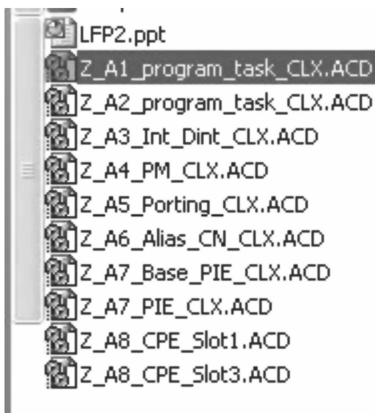


图 11-1　选择打开项目文件

三、使用多个周期性任务或一个周期性任务

在此部分中，将探索哪种编程方法更好：多个任务或带有多个程序的一个任务。

1. 在控制器项目管理器中，可注意到有一个连续任务和七个周期性任务。可能需要最小化任务文件夹。名为 Manager 的周期性任务用于处理其他任务属性。当前，该连续任务和五个周期性任务都已禁用，因此它们不会执行，如图 11-2 所示。

2. 在控制器项目管理器中，右键单击周期性任务 Periodic 并选择“属性”（Properties），如图 11-3 所示。

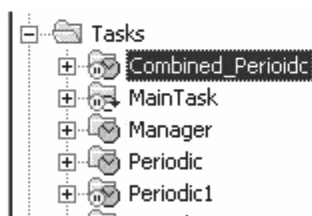


图 11-2　项目管理器列表

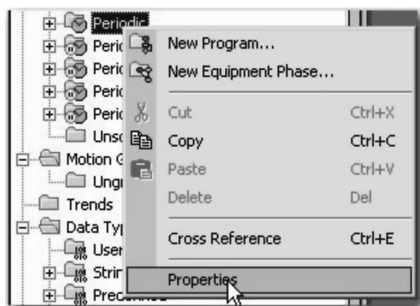


图 11-3　打开的任务菜单

3. 出现“任务属性”（Task Properties）窗口后，单击“监视”（Monitor）选项卡，如图 11-4 所示。

参考信息

任务属性监视选项卡

1）扫描时间（消逝时间）（Scan Times（Elapsed Time））：这将显示相应任务的最大消

逝时间和最后一次消逝时间。

2）间隔时间（触发之间的消逝时间）（Interval Times（Elapsed Time Between Triggers））：这将显示周期性任务触发之间的最大时间和最小时间。此时间值应约等于周期性任务的执行周期。

3）任务重叠计数（Task Overlap Count）：这是任务触发执行的次数，当前未显示任何值是因为它正在执行。当发生该情况时，控制器将记录一次次要故障。

4）重置（Reset）：此按钮可将所有字段都重置为零。

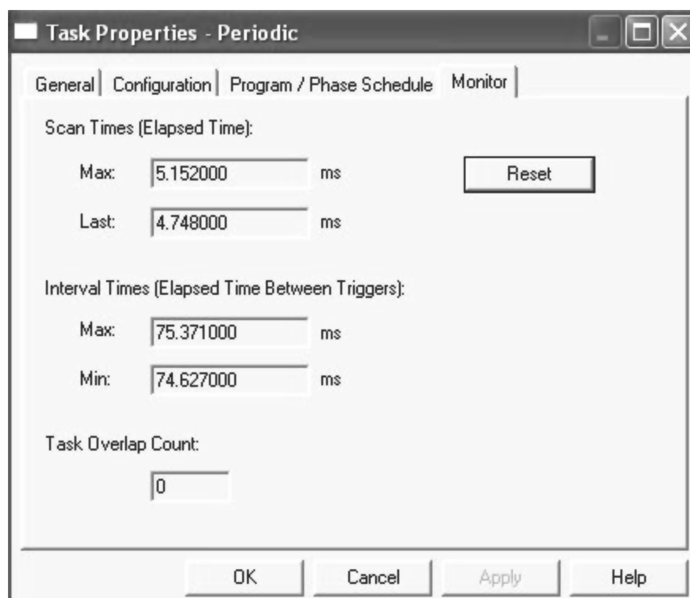


图 11-4　监视（Monitor）选项卡界面

4. 单击“重置”（Reset）按钮。

5. 可看到任务的扫描时间约为 5ms，间隔约为 75ms。任务的周期为 75ms。

6. 在控制器项目管理器中，双击“控制器标签”（Controller Tags）。这将打开标签监视器。在“标签监视器”（Tag Monitor）窗口中，能控制程序的执行方式。

7. 验证已处于“监视标签”（Monitor Tags）选项卡中。

8. 在标签监视器中，在 Inhibit_Periodic 的值列中输入 1 并按回车键。该值将恢复为零，如图 11-5 所示。

	Inhibit_Continuous	0
	Inhibit_Periodic	1
	Last_Perioidc	0

图 11-5　修改 Inhibit_Periodic 参数

9. 在标签监视器中，在 Uninhibit_Periodic 的值列中输入 1，并按回车键。该值将恢复为零，如图 11-6 所示。

这将导致其他已禁用的周期性任务开始执行。

	Uninhibit_Continuous	0
	Uninhibit_Periodic	1

图 11-6　修改 Uninhibit_Periodic 参数

10. 在“窗口”（Window）菜单中选择“任务属性-周期性”（Task Properties-

Periodic）。这会将“任务属性”（Task Properties）窗口置于顶层，如图 11-7 所示。

11. 出现“任务属性”（Task Properties）窗口后，单击“重置”（Reset）按钮。

12. 通过查看“任务属性”（Task Properties）窗口，将看到任务的扫描时间现在约为 21ms。这是因为在 5 个周期性任务之间进行了任务切换，如图 11-8 所示。

如果将周期性任务的周期降低到 22ms 会怎样？

13. 在“窗口”（Window）菜单中选择“控制器标签 s-A1_program_task（控制器）”（Controller Tags-A1_program_task（controller）），如图 11-9 所示。

这会将标签监视器窗口置于顶层。

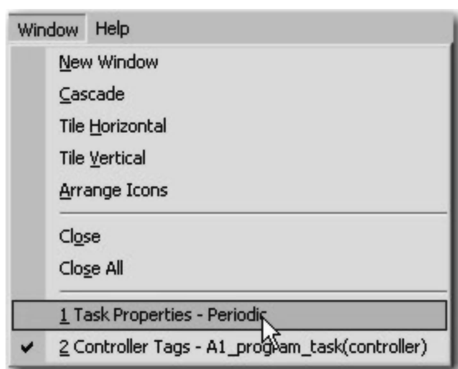


图 11-7　打开的窗口菜单

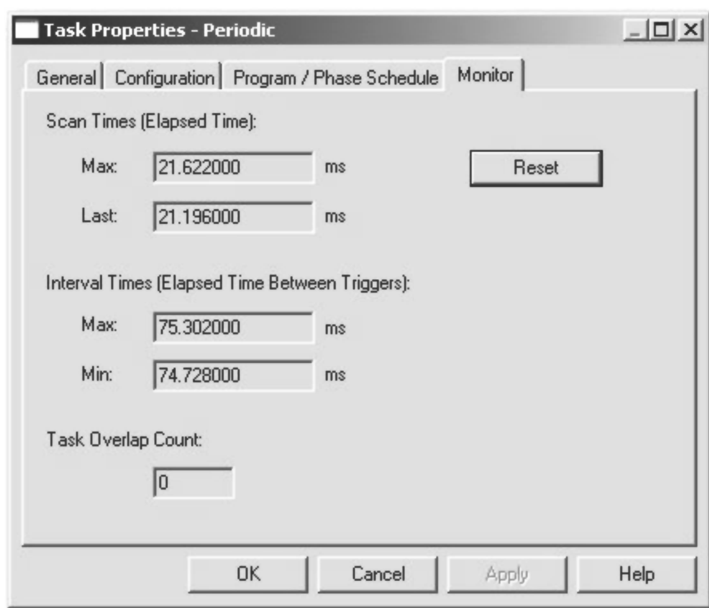


图 11-8　打开的任务属性窗口

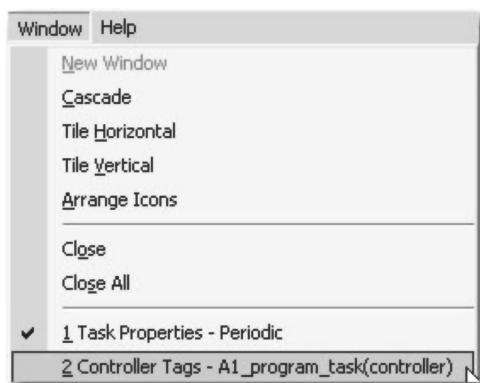


图 11-9　打开的窗口菜单

14. 在标签监视器中，在 Periodic_to_22 的值列中输入 1，并按回车键。该值将恢复为零。这样会将所有周期性任务的周期更改为 22ms，如图 11-10 所示。



图 11-10 修改 Periodic_to_22 的值

15. 将“任务属性-周期性”（Task Properties-Periodic）窗口置于顶层。

16. 单击“重置”（Reset）按钮，如图 11-11 所示。可看到，该任务的扫描时间仍约为 21ms，但现在间隔时间的最大值显示为约 24ms。最好是第一次便采用此周期性任务速率，是这样吗？

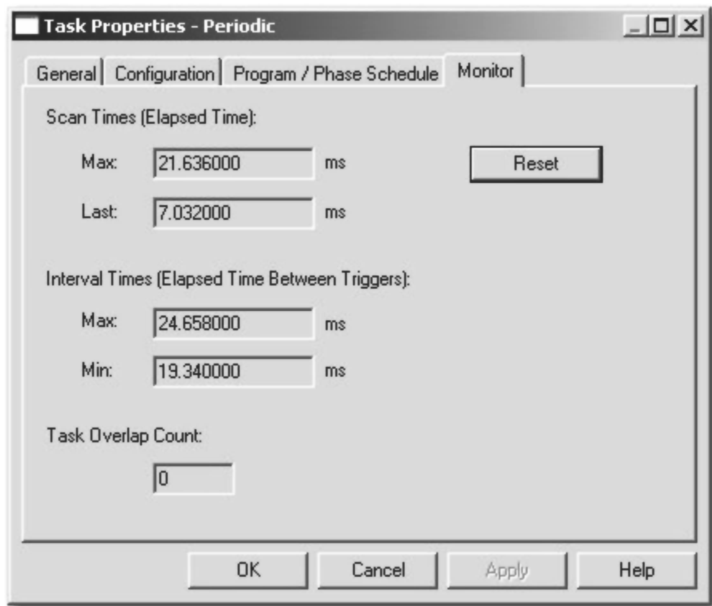


图 11-11 打开的任务属性窗口

17. 在控制器项目管理器中，右键单击周期性任务 Periodic4，并选择“属性”（Properties），如图 11-12 所示。

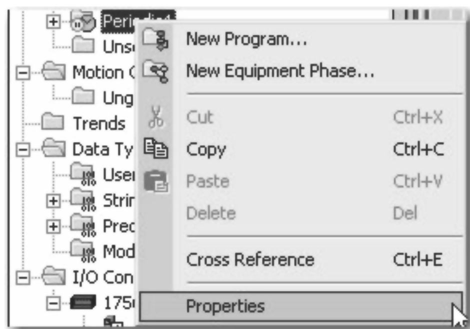


图 11-12 项目任务菜单

18. 出现“任务属性-Periodic4”（Task Properties-Periodic4）窗口后，单击“监视”（Monitor）选项卡。

19. 出现“监视”（Monitor）选项卡后，单击“重置”（Reset）按钮，如图 11-13 所示。在此处看到的情况会出乎大多数人的预料。

会发生任务重叠，而且扫描时间不是 21ms。这都是因为进行了任务切换。

请记住周期性任务每隔 1ms 将进行一次任务切换。使用时间线可轻松地显示这一点。例如，在下面：

|P1 P2 P3 P4 P5|P1 P2 P3 P4 P5|P1 P2 P3 P4 P5|P1 P2 P3 P4 P5|P1 P2 P3 P4 P5|

每次出现 P1 就表示周期任务 1 执行了 1ms。

那么周期性任务 P1 将执行多长时间呢？数一下查到第五个 P1 时共有多少个 PX。共有 21 个，

即需要 21ms，这就是看到的情况。那么，周期性任务 P5 需要多长的执行时间？通过计数将得到 25ms，这就是 Periodic5。这和之前看到的情况一致。

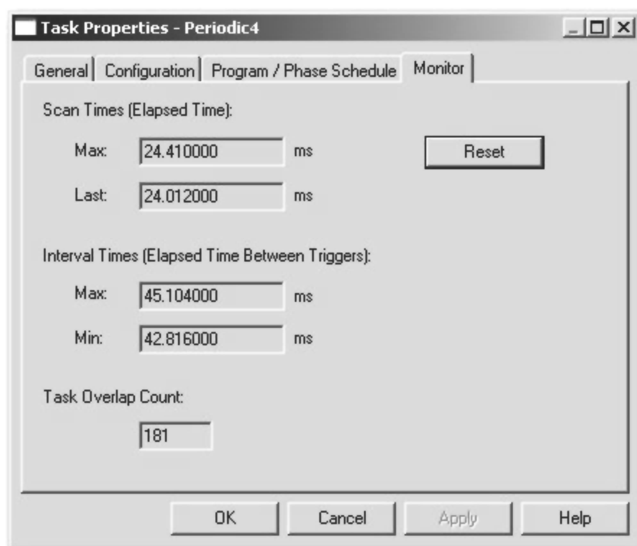


图 11-13　打开的任务属性窗口

20. 返回到“标签监视器”（Tag Monitor）并在 Periodic_to_75 的值列中输入 1。该值将恢复为零。这样会将所有周期性任务的周期更改回 75ms。

因此，按 22ms 来运行周期性任务并不是一个好主意。可认为，“我将按 26ms 来运行它们，这样就有一定的裕量，也就没有问题了。”

21. 将标签 Task_22 的值修改为 26000，如图 11-14 所示。这样便会将该周期性任务的周期更改为 26ms。

22. 在 Periodic_to_22 的值列中输入 1，并按回车键。该值将恢复为零。这样会将所有周期性任务的周期更改回 26ms。

+-Task_22	26000
+-Task_75	75000
Uninhibit_Continuous	0

图 11-14　周期任务属性

23. 切换回“任务属性-周期性”（Task Properties-Periodic）窗口。

24. 单击“重置”（Reset）按钮。看起来一切都好，但 Periodic4 却需要关注一下。

25. 切换到“任务属性-Periodic4”（Task Properties-Periodic4）窗口。

26. 单击“重置”（Reset）按钮。发现没有任何任务重叠发生，扫描时间约为 24ms。一切正常，任务在预期的时间内执行，而且未发生任何重叠。

27. 单击“确定”（OK），关闭两个“任务属性”（Task Properties）窗口。

28. 在“标签监视器”（Tag Monitor）窗口中，在 Inhibit_Periodic 的值列中输入 1 并按回车键。该值将恢复为零。刚刚禁用了所有周期性任务的执行。

29. 在“标签监视器”（Tag Monitor）窗口中，在 Combined_Periodic 的值列中输入 1 并按回车键。该值将恢复为零。刚刚开始执行 Combined_Periodic 任务。

30. 在控制器项目管理器中，右键单击周期性任务 Combined_Periodic 并选择“属性”（Properties）。

31. 出现“任务属性”（Task Properties）窗口后，单击“组态”（Configuration）选项卡。将

注意到周期已设置为25ms。

32. 单击“监视”（Monitor）选项卡并单击“重置”（Reset）按钮，如图 11-15 所示。

可看到执行时间短于25ms，而且未发生任何重叠，但是当具有 5 个单独的周期性任务时，便会发生重叠，这时便必须将周期时间设置为26ms。在此情况下，程序不会有额外的任务切换开销。

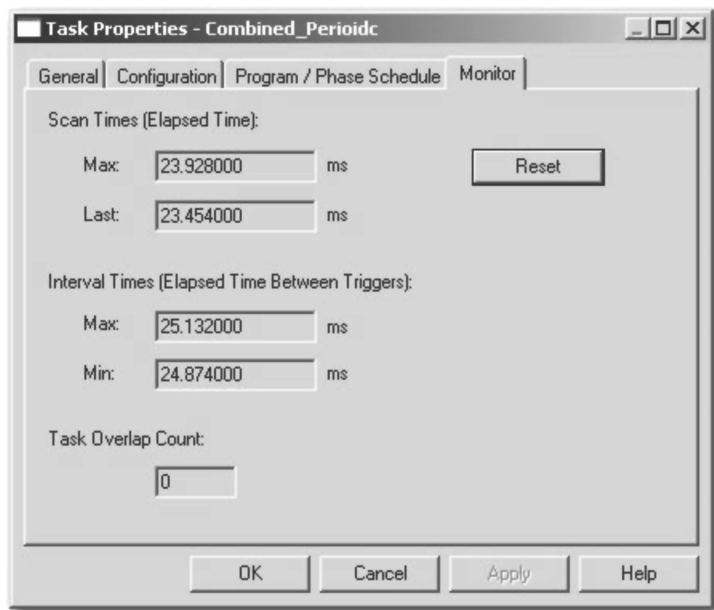


图 11-15 打开的任务属性窗口

可看到，与之前相比扫描的时间小于周期性任务 Periodic4 的时间。使用 Combined_ Periodic 任务可获得更好的性能。

33. 单击“确定”（OK），关闭“任务属性”（Task Properties）窗口。

34. 在“通信”（Communication）菜单中选择“脱机”（Go Offline），如图 11-16 所示。

35. 在工具栏菜单中，选择“文件 > 退出”（File > Exit），退出并且不保存所做的更改。

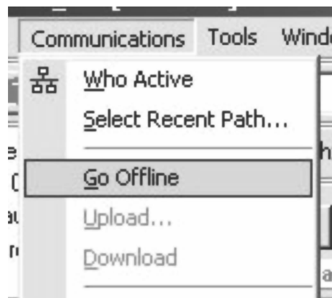


图 11-16 通信菜单

四、实验小结

在本实验中，介绍了通过一个任务中的多个程序进行程序分段要优于使用多个任务。这样性能会更好，而且不必担忧任务切换问题。

另外，也希望避免将多个任务设置为相同的优先级，避免采用类似的周期时间，以防止由于任务切换而增加扫描时间。

示例程序的所有周期性任务都采用了相同的周期时间和优先级。如果项目有如下规划，那么应怎样呢?

1）Periodic1 任务周期 2ms，优先级 1，执行时间 1ms；

2）Periodic2 任务周期 4ms，优先级 2，执行时间 0. 5ms；

3）Periodic3 任务周期 5ms，优先级 2，执行时间 2ms；

4）Periodic4 任务周期 15ms，优先级 7，执行时间 4ms；

5）Periodic5 任务周期 20ms，优先级 10，执行时间 9ms。

该程序的更好规划如下：

1）Periodic1 任务周期 2ms，优先级 1，执行时间 1ms；

2）Periodic23 任务周期 4ms，优先级 2，执行时间 2.0ms（将 Periodic2 和 Periodic3 组合到一起）；

3）Periodic45 任务周期 15ms，优先级 7，执行时间 11ms（将 Periodic4 和 Periodic5 组合到一起）。

这里通过组合的方式避免了多个任务具有相同的优先级，也避免了多个任务按类似的周期运行。

实验 12：理解周期性任务对连续任务的影响

一、实验目的

在这部分实验中，将了解以下概念：

周期性任务对连续任务的影响，将使用 Logix5000 任务监视器工具来分析此项目中连续任务和周期性任务之间的交互。该工具随 RSLogix5000 的提供，可用于监视在线项目中任务的运行情况。

在传统 PLC 中，有三种不同类型的“任务”：

1）主任务 = Logix 中的连续任务；

2）STI 任务 = Logix 中的周期性任务；

3）PII/DII 任务 = Logix 中的事件任务。

但是，只能有一个 STI 和 PII/DII。在 Logix 中可以有多个周期性任务和事件任务。这可以是一个功能强大的工具，但在执行前应先了解它。

二、打开现有控制器文件并将其下载到控制器中

在实验的该部分中，将打开一个现有程序。

1. 在计算机桌面上，双击 Lab Files 文件夹。

2. 双击 Z_A2_program_task_CLX. ACD 文件以在 RSLogix 5000 中启动该项目，如图 12-1 所示。

3. 在工具栏菜单中，选择“通信 > 在线节点”（Communication > Who Active）。

4. 浏览到 ControlLogix 演示箱插槽 1 中的控制器（请注意站上 1756-ENBT 模块的 IP 地址 10. 112. 2. 1x3，其中 x 是学员站的编号）。

5. 单击“下载”（Download）转到“远程运行”（Remote Run）模式。

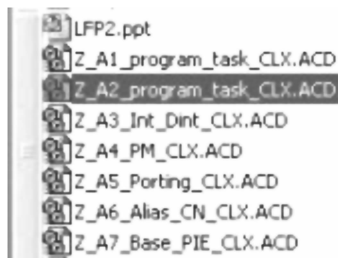


图 12-1　项目列表

三、理解周期性任务对连续任务的影响

在本实验中，将了解周期性任务对连续任务的执行所产生的影响。

在控制器项目管理器中，可注意到有一个连续任务和七个周期性任务。可能需要最小化任务文件夹。名为 Manager 的周期性任务用于处理其他任务属性。当前，该连续任务和 5 个周期性任务都已禁用，因此它们不会执行。

1. 在控制器项目管理器中，双击“控制器标签”（Controller Tags）。这将打开标签监视器。在“标签监视器”（Tag Monitor）窗口中，能控制程序的执行方式。

2. 验证已处于“监视标签”（Monitor Tags）选项卡中。

3. 在“标签监视器”（Tag Monitor）窗口中，在 Inhibit_Periodic 的值列中输入 1 并按回车键。这将禁用周期性任务。

4. 在“标签监视器”（Tag Monitor）窗口中，在 Uninhibit_Continuous 的值列中输入 1 并按回车键。这将禁用 Combined_Periodic 任务并令连续任务开始执行。

5. 在控制器项目管理器中，右键单击 MainTask 并选择“属性”（Properties）。

6. 出现“任务属性-主任务”（Task Properties-MainTask）窗口后，单击“监视”（Monitor）选项卡，如图 12-2 所示。

7. 可看到，连续任务的扫描时间约为 23. 872ms 到 25. 862ms（见图 12-2）。

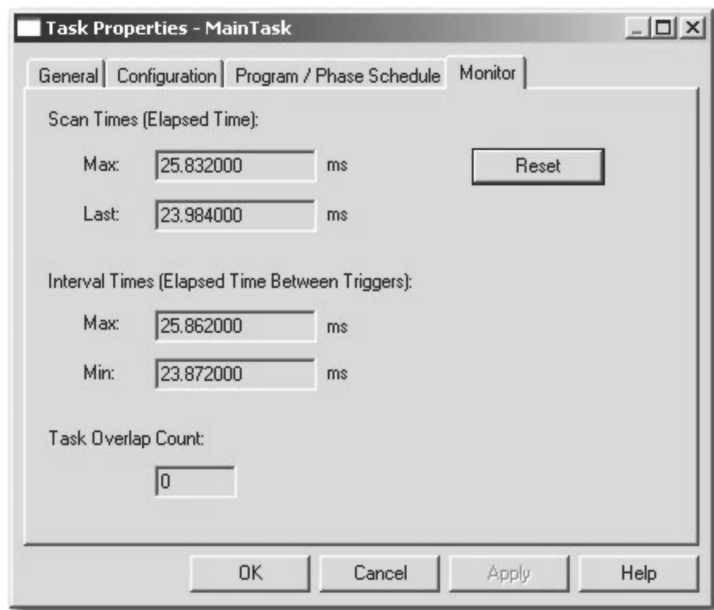


图 12-2　打开的任务属性窗口

现在将打开 Logix5000 任务监视器工具，以同时监视所有项目任务。

8. 在计算机桌面上，双击 Logix5000 任务监视器图标。

9. 在工具栏菜单中，选择“通信 > 监视”（Communications > Monitor），如图 12-3 所示。

10. 出现 RSWho 窗口后，左键单击以太网驱动程序前面的 + 符号展开其中的选项，如图 12-4 所示。

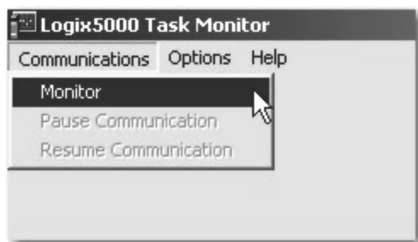


图 12-3　任务监视器通信菜单

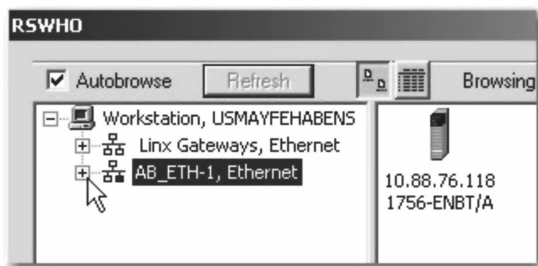


图 12-4　监控对象选择列表

11. 出现以下界面后，左键单击演示箱中 1756-ENBT 的 IP 地址前面的 + 符号，以展开其中的选项（10. 112. 2. 1x3，其中 x 是学员站的编号），如图 12-5 所示。

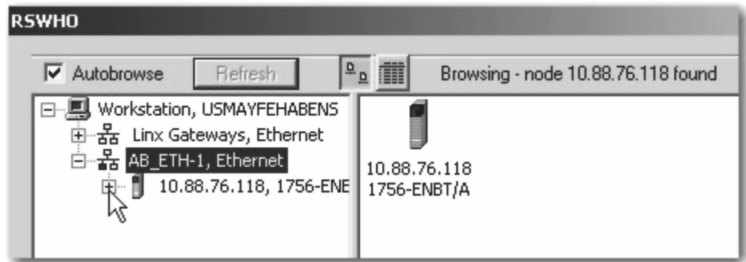
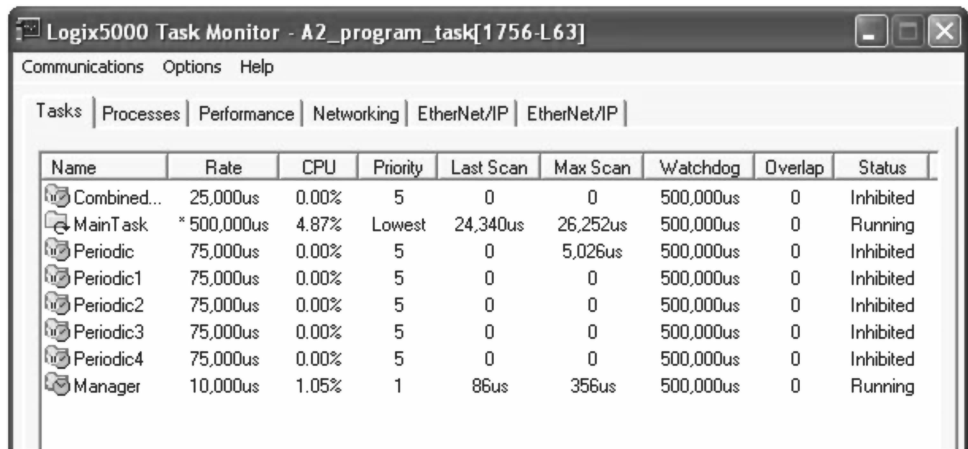


图 12-5 选择监控对象

12. 在 ControlLogix 演示箱插槽 1 中选择 1756-L63 并单击“确定”（OK）。如图 12-6 所示。



Name	Rate	CPU	Priority	Last Scan	Max Scan	Watchdog	Overlap	Status
Combined...	25,000us	0.00%	5	0	0	500,000us	0	Inhibited
MainTask	*500,000us	4.87%	Lowest	24,340us	26,252us	500,000us	0	Running
Periodic	75,000us	0.00%	5	0	5,026us	500,000us	0	Inhibited
Periodic1	75,000us	0.00%	5	0	0	500,000us	0	Inhibited
Periodic2	75,000us	0.00%	5	0	0	500,000us	0	Inhibited
Periodic3	75,000us	0.00%	5	0	0	500,000us	0	Inhibited
Periodic4	75,000us	0.00%	5	0	0	500,000us	0	Inhibited
Manager	10,000us	1.05%	1	86us	356us	500,000us	0	Running

图 12-6 选中的监控任务列表

参考信息

Logix5000 任务监视器任务选项卡概述

Logix5000 任务监视器为连接的控制器显示有关 CPU 分配的控制器应用信息。

“任务”（Tasks）选项卡显示与控制器内各个任务有关的信息。每个任务都具有一个指示任务类型及其状态（即，已禁用或未禁用）的图标。

参数：

1）名称（Name）：任务的名称，以及表示任务类型的图标。

2）速率（Rate）：计划的任务更新时间间隔（以毫秒为单位）。

3）CPU：每个任务的处理器占用百分比。

4）优先级（Priority）：获取执行所需资源的任务相对优先级。任务的优先级数越低，代表的优先级越高。

5）上次扫描时间（Last Scan）：任务从开始执行到结束已耗用的时间（以毫秒为单位）。

6）最长扫描时间（Max Scan）：任务从开始执行到结束所需的最长时间。

7）看门狗（Watchdog）：为任务分配的最长执行时间。

8）状态（Status）：任务的内部状态。有效的状态包括：

a）已停止（Stopped）：任务未运行，但也未禁用。

b）运行中（Running）：任务当前正在进行扫描和执行。

c）已禁用（Inhibited）：任务已被禁止执行（任务已禁用）。

9）“刷新”（Refresh）按钮：通过“刷新”（Refresh）按钮可刷新任务名称。如果已添加任务或已对其名称进行编辑，则需要单击“刷新”（Refresh）按钮来刷新名称列表。

10）日志（Log）：选中后将启用对此选项卡的记录。与此选项卡相关的数据将按照 CSV 的格式记录下来，并将以此文件名保存在 C：盘的根目录下：

C:\[controller name]_Tasks[mmddyyy].csv

软件将按照组态的更新速率（默认值：约 1000ms）向此文件中记录。可更改记录速率，具体方式是选择“选项- >通信更新速率”（Options- >Communication Update Rate）并设置一个介于 200ms 和 5000ms 之间的值。

11）“重置最长扫描时间”（Reset Max Scan）按钮：单击此按钮可重置每个任务的“最长扫描时间”（Max Scan Time），从而可提供更精确的最长扫描时间值。任务列表中至少有一项突出显示时，才可启用该按钮。右键单击列表中的项时，将出现一个弹出菜单，可用于重置扫描时间。只有在任务列表中有突出显示的项时，此弹出菜单才会出现。允许在列表中突出显示多个任务，可以同时重置所有这些任务的最长扫描时间。

12）“重置重叠计数”（Reset Overlap Count）按钮：单击此按钮将所选任务的重叠计数设置为零。任务列表中至少有一项突出显示时，才可启用该按钮。同样，右键单击列表中的项时，将出现一个弹出菜单，可用于重置重叠计数。只有在任务列表中有突出显示的项时，此弹出菜单才会出现。允许在列表中突出显示多个任务，可以同时重置所有这些任务的重叠计数。可注意到一些“任务”（Task）文件夹上有一个符号。该暂停符号表示任务当前已禁用或未执行。

在任务监视器界面中可看到有两个任务正在运行，六个已禁用。还可在一个界面中查看这两个运行中任务的最大扫描时间和上次扫描时间。

13. 单击工具栏上的最小化按钮最小化“任务监视器”界面，如图 12-7 所示。

ask[1756-L63]

EtherNet/IP

ity	Last Scan	Max Scan	Watchdog	Overlap	Status
	0	0	500,000us	0	Inhibited
est	24,344us	27,250us	500,000us	0	Running
	0	5,182us	500,000us	0	Inhibited
	0	0	500,000us	0	Inhibited
	0	0	500,000us	0	Inhibited
	0	0	500,000us	0	Inhibited
	0	0	500,000us	0	Inhibited
	92us	492us	500,000us	0	Running

图 12-7　任务监视器运行截图

来看一下周期性任务对连续任务的影响。

14. 单击“确定”（OK），关闭“任务属性”（Task Properties）窗口。

15. 转到“标签监视器”(Tag Monitor)窗口。

16. 将标签 Task_22 的值修改为 25000 并按回车键。这样便可将周期性任务的周期更改为 25ms。

17. 在 Periodic_to_22 的值列中输入 1 并按回车键。该值将恢复为零。这样会将所有周期性任务的周期更改为 25ms。

18. 在 Uninhibit_Periodic 的值列中输入 1 并按回车键。该值将恢复为零。刚刚激活了所有周期性任务。

19. 通过查看控制器面板可注意到，控制器已发生故障，如图 12-8 所示。

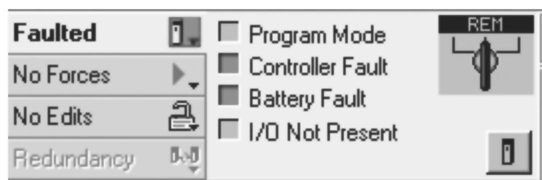


图 12-8 控制器面板

20. 在界面底部的工具栏上单击 Logix5000 任务监视器将其最大化，如图 12-9 所示。

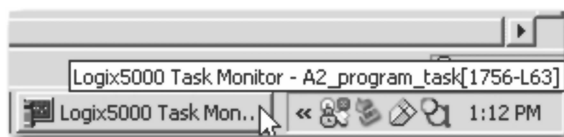


图 12-9 工具栏上的 Logix5000 任务监视器窗口

注意，该两个原本处于运行中状态的任务现已停止，因为控制器发生故障，如图 12-10 所示。

Logix5000 Task Monitor - A2_program_task[1756-L63]

Communications Options Help

Tasks | Processes | Performance | Networking | EtherNet/IP | EtherNet/IP

Name	Rate	CPU	Priority	Last Scan	Max Scan	Watchdog	Overlap	Status
Combined...	25,000us	0.00%	5	0	0	500,000us	0	Inhibited
MainTask	* 500,000us	0.00%	Lowest	0	500,016us	500,000us	0	Stopped
Periodic	25,000us	0.00%	5	0	21,246us	500,000us	0	Stopped
Periodic1	25,000us	0.00%	5	0	22,068us	500,000us	0	Stopped
Periodic2	25,000us	0.00%	5	0	22,690us	500,000us	0	Stopped
Periodic3	25,000us	0.00%	5	0	23,342us	500,000us	0	Stopped
Periodic4	25,000us	0.00%	5	0	24,048us	500,000us	0	Stopped
Manager	10,000us	0.00%	1	0	356us	500,000us	0	Stopped

图 12-10 监控任务列表

21. 最小化 Logix5000 任务监视器并单击 RSLogix5000 屏幕中文字“故障”(Faulted)旁边的“控制器”(Controller)图标。将出现一个下拉菜单。选择“转到故障部分”(Go To Faults)，如图 12-11 所示。

22. 将出现“控制器属性-A2_program_task”(Controller Properties-A2_ program_ task)窗口。发生的错误是“主任务”(MainTask)(连续任务)的“看门狗故障”(Watchdog Fault)，如图 12-12 所示。这意味着连续任务未能在 500ms 内完成执行过程。发生这一情况的原因是因为周期性任务耗用了连续任务的执行时间。

23. 单击“清除主要故障”(Clear Majors)按钮。

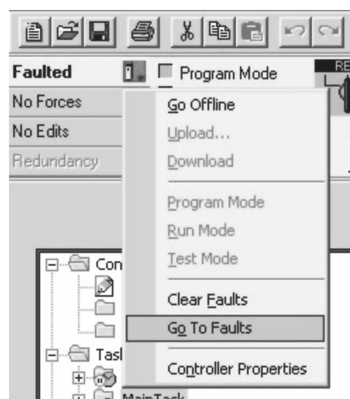


图 12-11 故障菜单

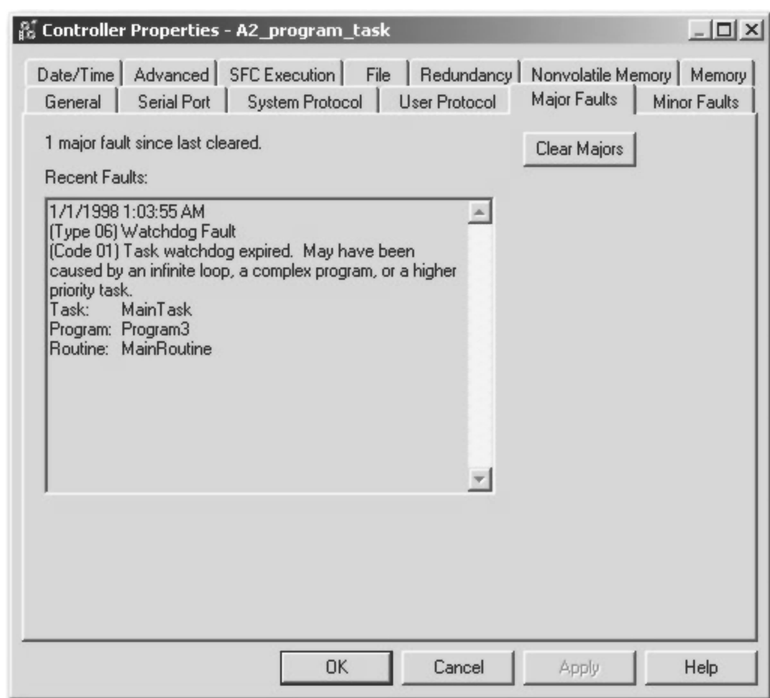


图 12-12 控制器属性界面

24. 单击“确定”（OK），关闭“控制器属性”（Controller Properties）窗口。需要修改周期性任务的周期，以防止“看门狗故障”的再次发生。

25. 在“标签监视器”（Tag Monitor）窗口中，在 Task_22 的值列中输入 27000 并按回车键。这会将周期任务周期更改为 27ms。

26. 在“标签监视器”（Tag Monitor）窗口中，在 Periodic_to_22 的值列中输入 1 并按回车键。这会令对周期进行的更改生效。

27. 将控制器设为运行模式。Periodic_to_22 的值将恢复为零。此时控制器未发生故障。

28. 打开“主任务”（MainTask）的“属性”（Properties）窗口。

29. 出现“任务属性-主任务”（Task Properties-MainTask）窗口后，单击“监视”（Monitor）选项卡。

30. 单击“重置”（Reset）按钮，如图 12-13 所示。

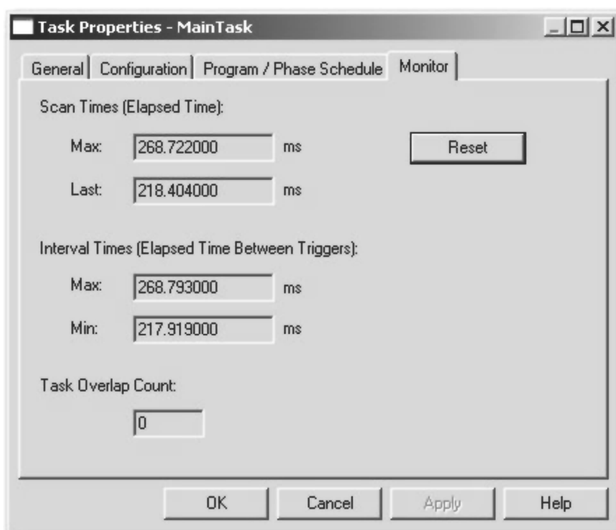


图 12-13 打开的任务属性窗口

可注意到，该任务的扫描时间已大幅增加。在之前情况下，连续任务是唯一的任务，执行的时间为 217.919 ~ 268.739ms。发生这种情况的原因是周期性任务的优先级要高于

连续任务。

31. 单击“确定”（OK），关闭“任务属性”（Task Properties）窗口。

32. 在界面底部的工具栏上单击 Logix5000 任务监视器，将其最大化。Logix5000 任务监视器将显示如图 12-14 所示界面。

Logix5000 Task Monitor - A2_program_task[1756-L63]

Communications Options Help

Tasks | Processes | Performance | Networking | EtherNet/IP | EtherNet/IP

Name	Rate	CPU	Priority	Last Scan	Max Scan	Watchdog	Overlap	Status
Combined...	25,000us	0.00%	5	0	0	500,000us	0	Inhibited
MainTask	*500,000us	48.73%	Lowest	244,348us	269,724us	500,000us	0	Running
Periodic	27,000us	17.60%	5	21,172us	21,636us	500,000us	0	Running
Periodic1	27,000us	17.37%	5	22,162us	22,450us	500,000us	0	Running
Periodic2	27,000us	17.22%	5	22,432us	23,058us	500,000us	0	Running
Periodic3	27,000us	17.15%	5	23,222us	23,700us	500,000us	0	Running
Periodic4	27,000us	17.30%	5	23,936us	24,310us	500,000us	0	Running
Manager	10,000us	1.07%	1	82us	356us	500,000us	0	Running

图 12-14 监控任务列表

可注意到，Logix5000 任务监视器显示连续任务和 6 个正在执行的周期性任务，如图 12-15 所示。连续任务的“最长扫描时间”（Max Scan）值（在图 12-15 中突出显示）实际上显示已超出“看门狗”（Watchdog）值，从而导致控制器故障。

Logix5000 Task Monitor - A2_program_task[1756-L63]

Communications Options Help

Tasks | Processes | Performance | Networking | EtherNet/IP

Name	Rate	CPU	Priority	Last Scan	Max Scan	Watchdog	Overlap	Status
Combined...	25,000us	0.00%	5	0	0	500,000us	0	Inhibited
MainTask	*500,000us	53.62%	Lowest	268,110us	500,016us	500,000us	0	Running
Periodic	27,000us	17.54%	5	21,470us	21,800us	500,000us	0	Running
Periodic1	27,000us	17.19%	5	21,904us	22,494us	500,000us	0	Running
Periodic2	27,000us	17.26%	5	22,622us	23,140us	500,000us	0	Running

图 12-15 监控任务列表中导致故障的任务信息

33. 关闭并退出 Logix5000 任务监视器。

34. 在 RSLogix 5000 的工具栏菜单中，选择“通信 > 脱机”（Communication > Go Offline），如图 12-16 所示。

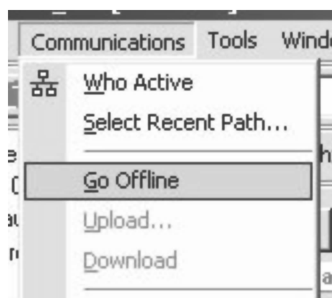


图 12-16 通信菜单

35. 在工具栏菜单中，选择“文件 > 退出”（File > Exit），退出并且不保存所做的更改。

四、实验小结

在本实验中，介绍了周期性任务的执行将对连续任务的执行造成的影响。对周期性任务与连续任务的需求进行平衡很重要。请记住，控制器中的通信是在连续任务执行过程中（在系统内务处理时间片过程中）提供服务的。因此，如果令连续任

务的执行时间不足，则控制器的通信也将受到影响。这将影响消息块传送、HMI 更新、RSLogix5000、控制器建立连接等。

使用 Logix5000 任务监视器工具（随 RSLogix5000 提供）来分析连续任务和周期性任务之间的交互，并用其来轻松查看项目中所有任务的所有参数。该工具是了解控制器的非常实用的在线窗口，可在一个界面中显示所有任务的运行情况。有关此实验材料中未包括的控制器在线操作的附加视图和信息，请参见“过程”（Processes）、“性能”（Performance）、“网络”（Networking）和“EtherNet I/P”选项卡选项。

实验 13：使用部分导入/导出工具

一、实验目的

在本实验中，将了解以下概念：

使用部分导入/导出工具来帮助加速程序开发。

二、关于梯形图部分导入/导出工具

大多数情况下，应用开发时都会有许多代码段重复许多次。例如，在传送带系统中具有与各个传送带关联的核心代码；启动/停止、报警和电源管理代码。

之前，通常可能会采用片段文件的方法，具体操作是剪切梯形图的梯级并将其保存为桌面上的片段文件，或者直接剪切/复制粘贴。但当进行此操作时，可能必须花时间来创建所有标签并更改标签名称。正期待着更方便快速的方法。

参考信息

梯形图部分导入/导出工具

这是 RSLogix 5000 版本 13 的新功能。

梯形图部分导入/导出工具提供的基本功能是从项目中提取特定梯级的代码并将其保存到外部文件中（L5X）。之后，可以将代码导入到任何 RSLogix 5000 项目中。其中一个重要区别是，部分导入/导出功能不仅限于诸如片段文件的梯形图梯级。导出文件包含所有的标签定义、UDT 和关联的注释。因此，导入文件后，所有创建工作便已完成，产生的代码不会验证出任何错误。

除此之外，RSLogix 5000 的功能还更进一步，它提供一个列出所有关联标签、UDT 和说明的对话框，从而可在导入之前进行相应的修改。这有助于避免潜在的标签命名冲突。

三、使用梯形图部分导入/导出工具导出到 L5X 文件

1. 在计算机桌面上，双击 Lab Files 文件夹。
2. 双击 LFP2 文件夹。
3. 双击 Z_A7_Base_PIE_CLX. ACD 文件，以在 RSLogix 5000 中启动该项目，如图 13-1 所示。
4. 在控制器项目管理器中，双击 Conveyor 文件夹以将其展开。这便是代码保存的位置。一旦 Conveyor 文件夹展开后，其类似图 13-2 所示的项目管理器列表。
5. 双击 Conveyor 例程，以打开梯形图编辑器。
6. 在工具栏菜单中，选择“编辑 > 全选”（Edit > Select All），如图 13-3 所示。这将选中此例程中的全部 8 个梯级。
7. 右键单击梯级 0 的左侧，并选择“导出梯级”（Export Rung），如图 13-4 所示。

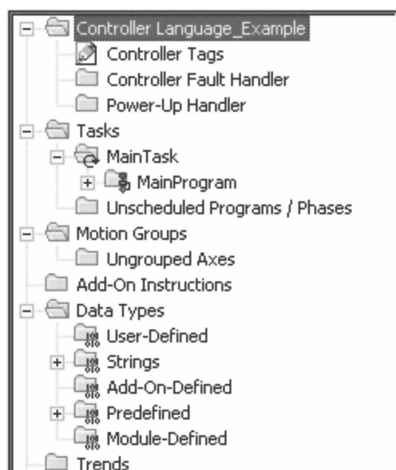


图 13-1 项目列表

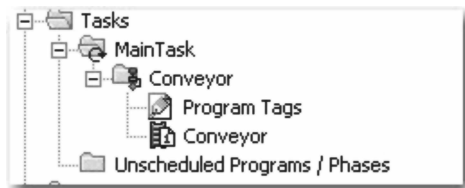


图 13-2 项目管理器列表

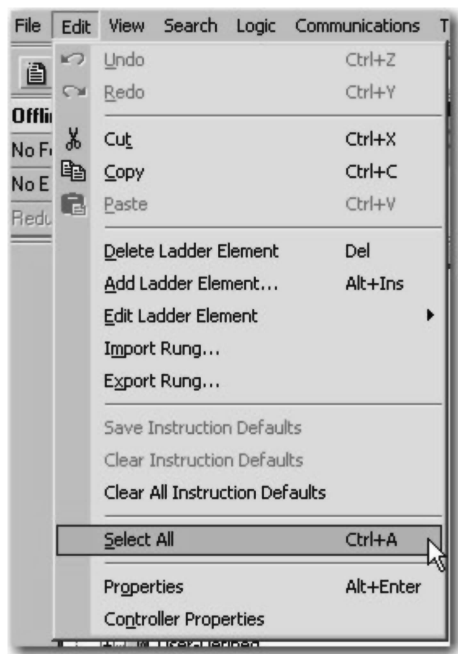


图 13-3 编辑菜单

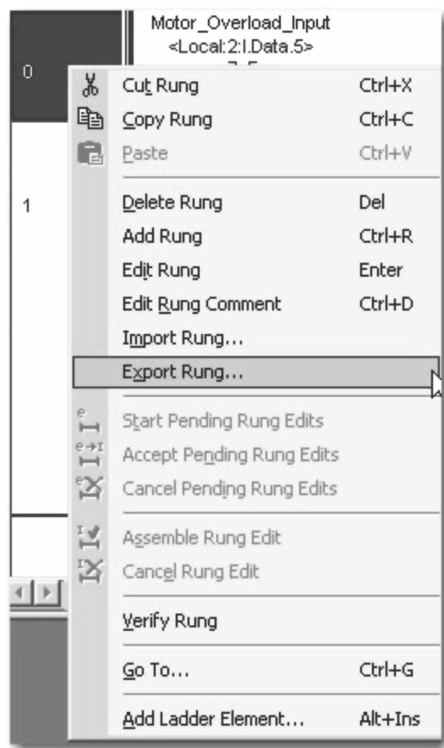


图 13-4 梯形图编辑菜单

8. 出现“导出梯级”（Export Rung）窗口后，在“文件名”（File name）字段中输入 Z_A7_Conveyor_Export，然后单击“导出”（Export），如图 13-5 所示。

导出的梯形图文件（L5X）已创建。

9. 在工具栏菜单中，选择“文件 > 退出”（File > Exit），退出并且不保存所做的更改。

四、使用梯形图部分导入/导出工具导入 L5X 文件

1. 在计算机桌面上，双击 Lab Files。

2. 双击 Z_A7_PIE_CLX. ACD 文件，在 RSLogix 5000 中打开该项目，如图 13-6 所示。

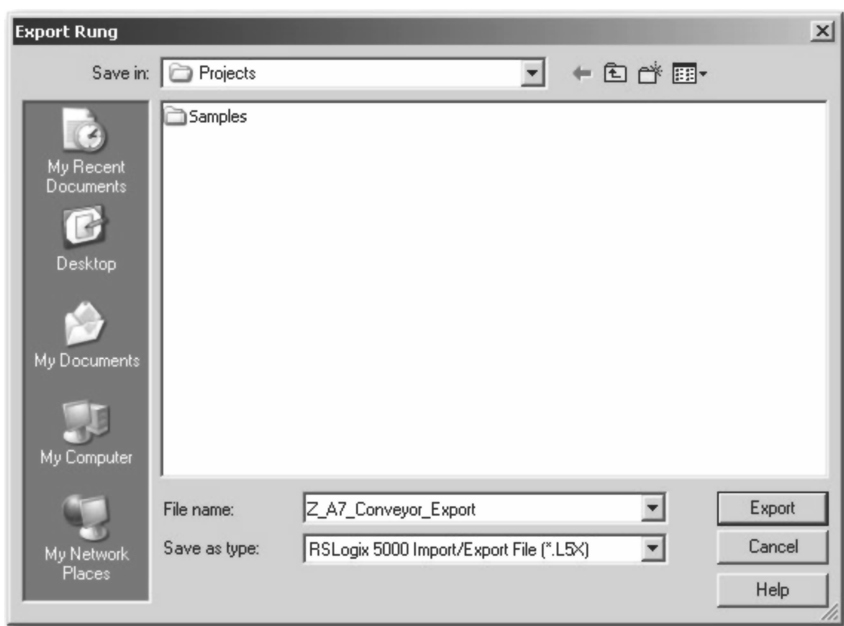


图 13-5　项目导出保存窗口

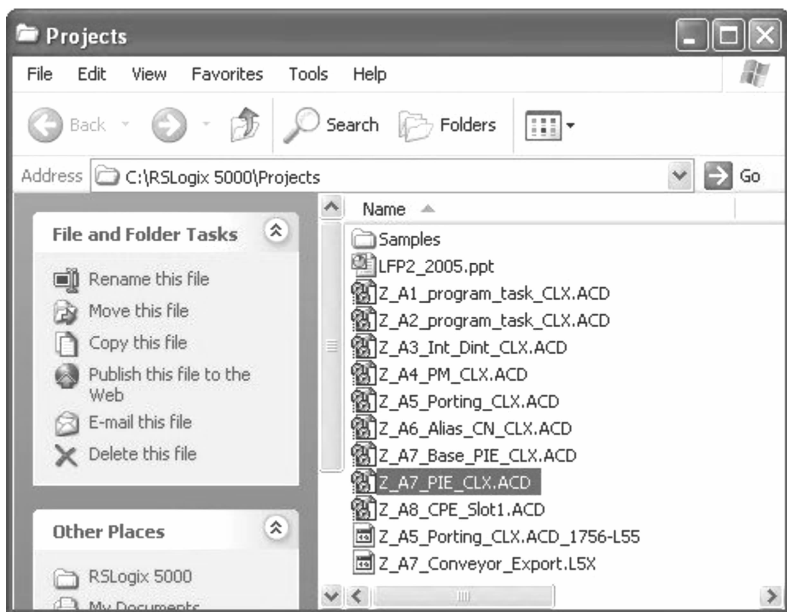


图 13-6　项目列表

3. 在控制器项目管理器中，双击 Conveyor 程序文件夹将其展开。将在此位置添加代码。

4. 双击 Conveyor 例程。将打开梯形图编辑器，该例程中没有任何梯级，如图 13-7 所示。

5. 在梯形图编辑器中，右键单击梯级 0 并选择“导入梯级”（Import Rung），如图 13-8 所示。

6. 将弹出“导入梯级”（Import Rung）窗口。单击文件 Z_A7_Conveyor_Export. L5X。然后单击“导入”（Import），如图 13-9 所示。

将弹出“导入组态”（Import Configuration）窗口，如图 13-10 所示。

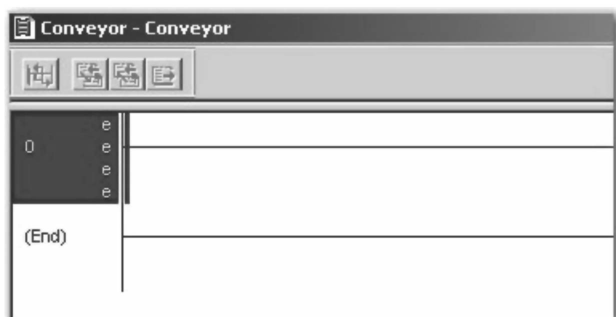


图 13-7　打开的空白梯形图

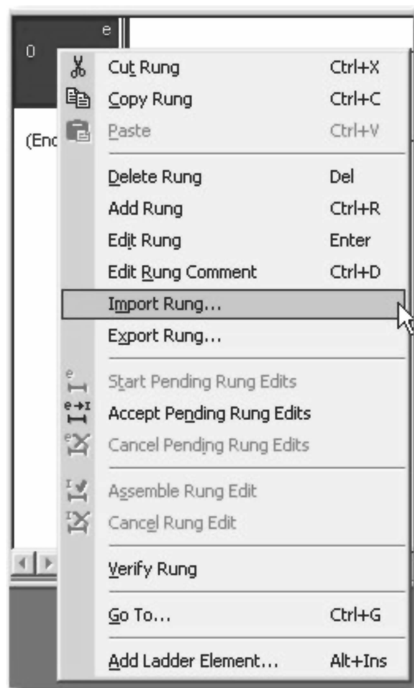


图 13-8　梯形图编辑菜单

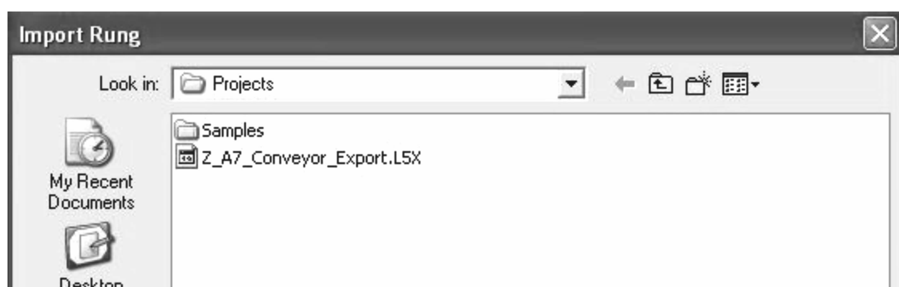
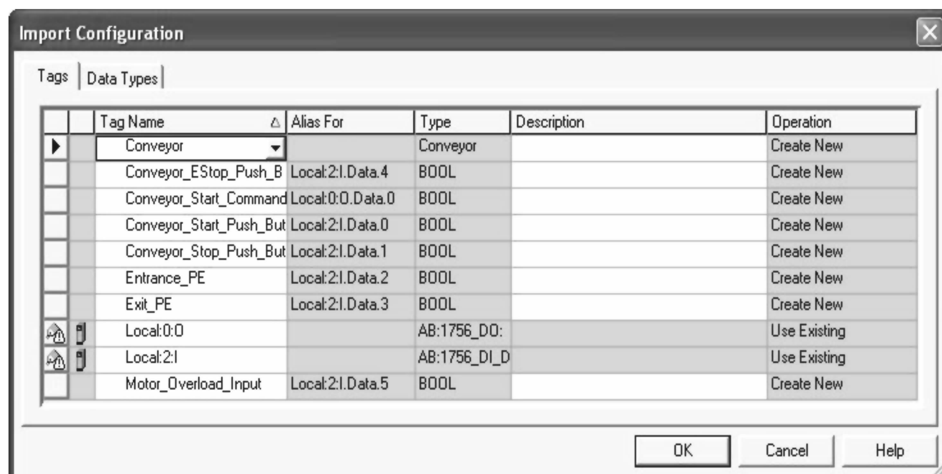


图 13-9　导入梯级窗口



Tag Name	Alias For	Type	Description	Operation
Conveyor		Conveyor		Create New
Conveyor_EStop_Push_B	Local:2:I.Data.4	BOOL		Create New
Conveyor_Start_Command	Local:0:O.Data.0	BOOL		Create New
Conveyor_Start_Push_But	Local:2:I.Data.0	BOOL		Create New
Conveyor_Stop_Push_But	Local:2:I.Data.1	BOOL		Create New
Entrance_PE	Local:2:I.Data.2	BOOL		Create New
Exit_PE	Local:2:I.Data.3	BOOL		Create New
Local:0:O		AB:1756_DO:		Use Existing
Local:2:I		AB:1756_DI_D		Use Existing
Motor_Overload_Input	Local:2:I.Data.5	BOOL		Create New

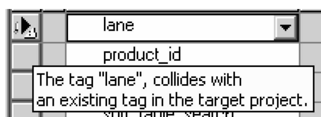
图 13-10　导入组态窗口

参考信息

导入组态

“导入组态”（Import Configuration）向导包含两个选项卡。

1）“标签”（Tags）：与要导入的梯级相关的所有标签均在此列出。在执行导入操作前，可以更改任何标签的名称。如果项目中已存在该标签名，系统将发出警告。



这样便可以更改标签的名称以免出现问题。

2）“数据类型”（Data Types）：此选项卡将显示导入时引入的所有 UDT。该选项卡将显示 UDT 是新创建的（如果尚不存在）还是使用已有的（已经存在）。此选项卡的信息仅供参考。

7. 可以看到，“导入”（Import）窗口中的 I/O 模块显示了警告，如图 13-11 所示。这些警告表明这些 I/O 模块已存在于项目中。在“操作”（Operation）列，可以看到有的单元格显示“使用现有”（Use Existing）；这表示导入时将使用项目中的现有模块。

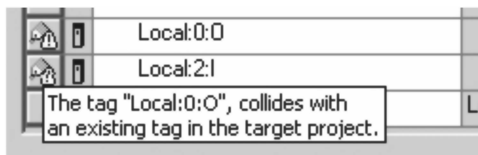


图 13-11 “导入”窗口的警告信息

最终需要添加两个传送带代码段。分别指定为 Conveyor1 和 Conveyor2。

8. 向标签中添加 1 来表示 Conveyor1。完成后，标签应显示如图 13-12 所示。

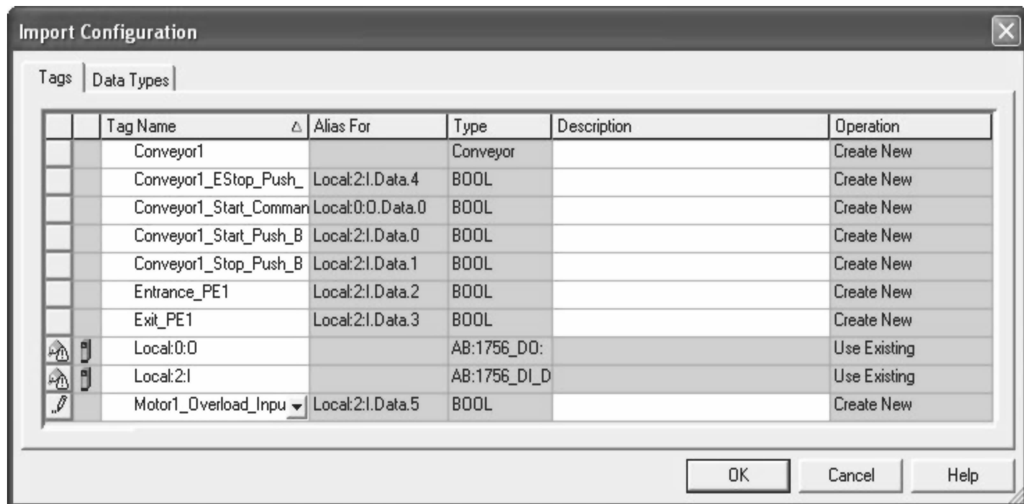


图 13-12 “导入”组态窗口

9. 单击“确定”（OK）。“导入”（Import）开始。完成后，便写好了 Conveyor1 的梯形图代码。所有标签都创建在合适的范围内；对于本例，该范围为 Conveyor。梯形图编辑器将显示如图 13-13 所示内容。

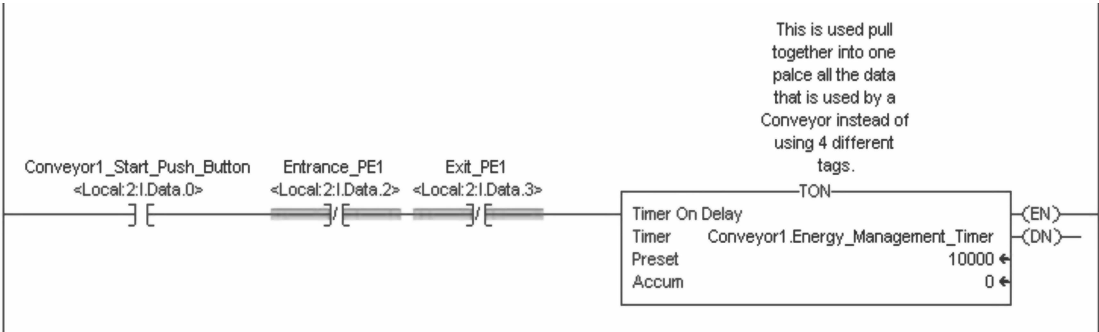


图 13-13 导入成功后的梯形图

10. 滚动到 Conveyor 例程顶部，并删除空的梯级 0。

11. 单击工具栏上的“校验控制器”（Verify Controller）图标。控制器验证后未发现错误。非常轻松地就添加了代码。

现在将添加 Conveyor2 的代码。但在开始之前，必须解决一个问题。按照 L5X 文件中代码的定义方式，该模块将在插槽 0 和 2 使用，并且将指定为特定点的别名。希望将 Conveyor2 设置为其他输入和输出点的别名。有以下三种方式可解决这一问题：

1）一种方法是按原样导入梯级并重新设置 Conveyor2 输入和输出的别名；

2）向项目中添加输入和输出模块，然后在导入过程中解决此问题；

3）修改 L5X 文件，重新将别名指定到现有模块的其他开放点上。

将执行后两个选项。

12. 首先执行第二个选项。只需复制现有 I/O 模块并将其粘贴回项目中。

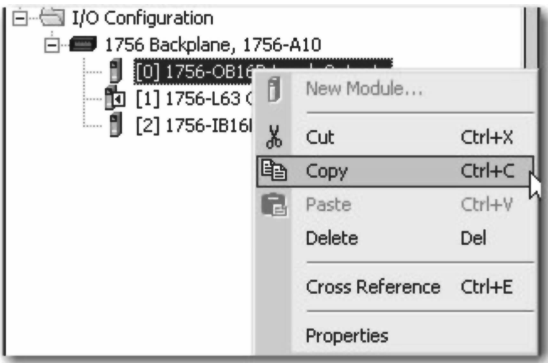


图 13-14 控制器项目管理器菜单

13. 在控制器项目管理器中，右键单击插槽 0 上的 1756-OB16D 模块，并单击“复制”（Copy），如图 13-14 所示。

14. 右键单击 I/O Configuration 文件夹，并单击“粘贴”（Paste），如图 13-15 所示。此操作可将新的 1756-OB16D 添加到插槽 3 的项目中。

15. 在插槽 2 上重复执行上述操作。

16. 检验 I/O 组态是否显示如图 13-16 所示。

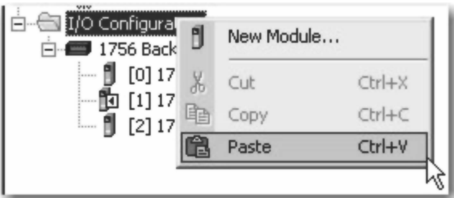


图 13-15 控制器项目管理器菜单

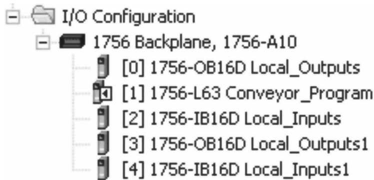


图 13-16 更新后的控制器项目列表

17. 现在可以导入 Conveyor2 的代码了。

18. 在梯形图编辑器中，右键单击梯级 0，并单击“导入梯级”（Import Rung），如图 13-17 所示。

19. 将弹出“导入梯级”（Import Rung）窗口。选择文件 Z_A7_Conveyor_Export. L5X。然后单击“导入”（Import）。

20. 弹出“导入组态”（Import Configuration）窗口后，更改标签的名称（方式与 Conveyor1 相同），如图 13-18 所示。首先需要在标签名称中添加“2”，就像此前添加 Conveyor1 那样。

21. 现在，要对 Local:0:O 和 Local:2:I 标签进行更改。希望输入指向在插槽 4 中添加的新输入模块，输出指向在插槽 3 中添加的输出模块。按照图 13-19 所示，更改“标签名称”（Tag Names）。

可以看到，当通过更改标签名称来反映新插槽位置时，“别名属于”（Alias For）一列中也会反映这些更改。

22. 单击“确定”（OK）。“导入”（Import）开始。完成后，便写好了 Conveyor2 的梯形图代码。所有标签都创建在合适的范围内；对于本例，该范围为 Conveyor。

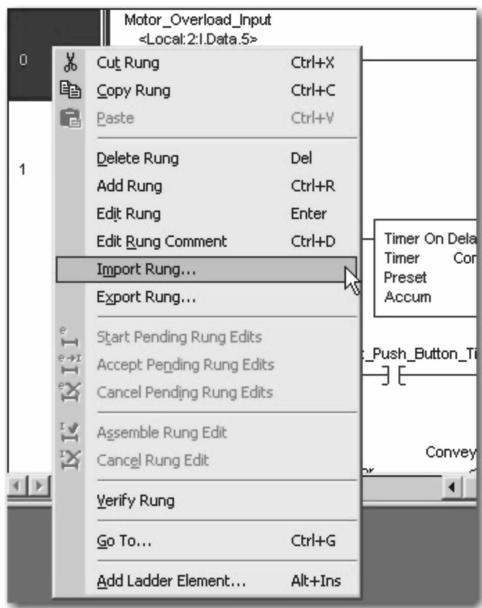
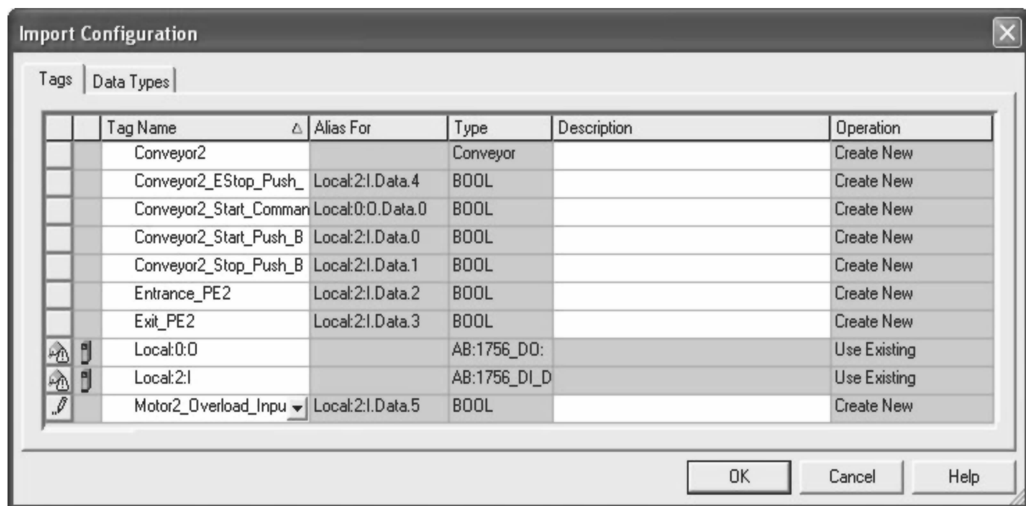


图 13-17 梯形图编辑器

23. 单击工具栏上的“校验控制器”（Verify Controller）图标。控制器验证后未发现错误。非常轻松地就添加了代码。



Tag Name	Alias For	Type	Description	Operation
Conveyor2		Conveyor		Create New
Conveyor2_EStop_Push_	Local:2:I.Data.4	BOOL		Create New
Conveyor2_Start_Comman	Local:0:O.Data.0	BOOL		Create New
Conveyor2_Start_Push_B	Local:2:I.Data.0	BOOL		Create New
Conveyor2_Stop_Push_B	Local:2:I.Data.1	BOOL		Create New
Entrance_PE2	Local:2:I.Data.2	BOOL		Create New
Exit_PE2	Local:2:I.Data.3	BOOL		Create New
Local:0:O		AB:1756_DO:		Use Existing
Local:2:I		AB:1756_DI_D		Use Existing
Motor2_Overload_Inpu	Local:2:I.Data.5	BOOL		Create New

图 13-18 “导入组态”窗口

24. 现在，将修改实际的 L5X 文件。在本例中，使用现有模块上未占用的输入和输出。将 Conveyor1 添加到系统时，会指定以下别名。

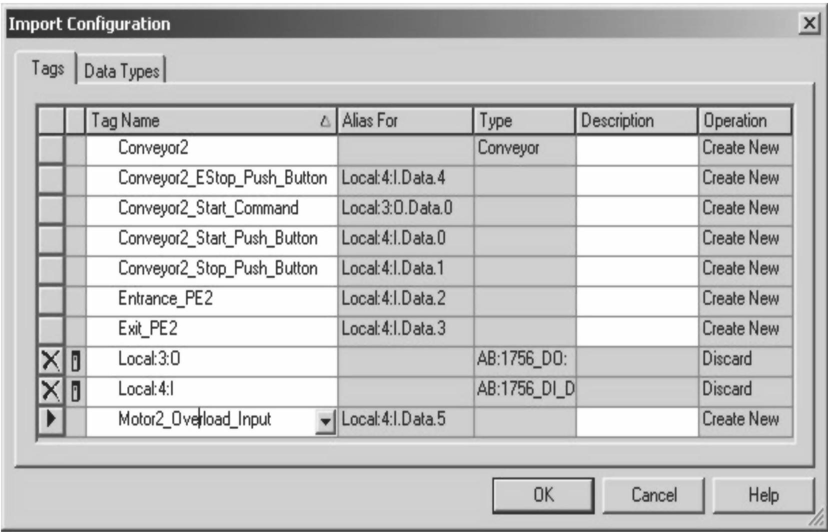


图 13-19 更改后的标签名称

对于插槽 2 上的 1756-IB16D：

1）点 0 = Conveyor_Start_Push_Button

2）点 1 = Conveyor_Stop_Push_Button

3）点 2 = Entrance_PE

4）点 3 = Exit_PE

5）点 4 = Conveyor_EStop_Push_Button

6）点 5 = Motor_Overload_Input

对于插槽 0 上的 1756-OB16D：

7）点 0 = Conveyor_Start_Command

对于新加入的传送带段，新的别名为“对于插槽 2 上的 1756-IB16D”：

8）点 6 = Conveyor_Start_Push_Button

9）点 7 = Conveyor_Stop_Push_Button

10）点 8 = Entrance_PE

11）点 9 = Exit_PE

12）点 10 = Conveyor_EStop_Push_Button

13）点 11 = Motor_Overload_Input

对于插槽 0 上的 1756-OB16D：

14）点 1 = Conveyor_Start_Command

现在，将使用其他程序而不是 RSLogix5000 来打开 L5X 文件。

25. 在计算机桌面上，双击 Lab Files 文件夹。

26. 双击 LFP2 文件夹。

27. 右键单击文件 Z_A7_Conveyor_Export. L5X，并选择“打开”（Open），如图 13-20 所示。

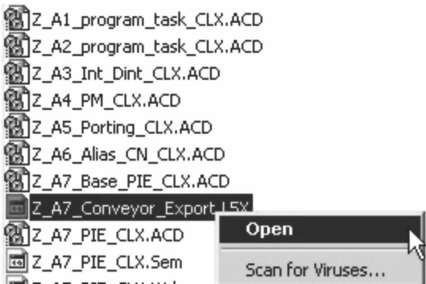


图 13-20 选择打开 L5X 文件

28. 当弹出以下窗口时，单击“从列表中选择程序”（Select the program from a list）单选按钮，如图 13-21 所示。

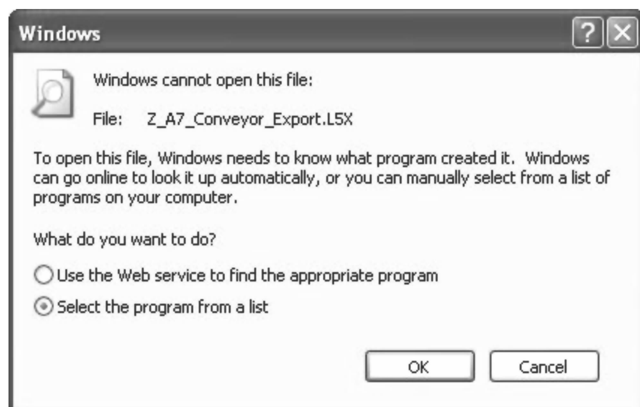


图 13-21　弹出的选择程序窗口

29. 向下滚动到“写字板”选项，并单击“确定”（OK），如图 13-22 所示。



图 13-22　选择写字板程序

30. 文件打开后将显示如下内容：


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RSLogix5000Content SchemaRevision="1.0" SoftwareRevision="13.03" Target
<Controller Use="Context" Name="Conveyor_Program" UId="f90ef335">
<DataTypes Use="Context" UId="d0c0ad05">
<DataType Name="Conveyor" Family="NoFamily" Class="User" UId="70111cf3">
<Description>
<![CDATA[This is used pull together into one palce all the data that is
</Description>
<Members>
```

31. 滚动浏览文件，找到“程序应用”（Program Use）部分。将在此处进行更改。

```
</Tags>
<Programs Use="Context" UId="3238c36d">
<Program Use="Context" Name="Conveyor" UId="31ee8d4e">
<Tags Use="Context" UId="e26b2fb2">
<Tag Name="Conveyor_Start_Command" UId="9c0a19c1" TagType
<Data>00</Data>
```

32. 找到以下行：

```
<Tag Name="Conveyor_Start_Command" UId="9c0a19c1" TagType="Alias" DataType="BOOL" Radix="Decimal" AliasFor="Local:0:
<Data>00</Data>
```

首个标签是 Conveyor_Start_Command。该行的后面可看到 AliasFor = "Local:0:O. Data. 0"。前面说过，现在希望将其设置为输出模块中点 1 的别名。

33. 按照下图更改 AliasFor 段：

```
AliasFor="Local:0:O.Data.1">
```

34. 现在继续浏览文件并根据前文第 24 步中列出的内容更改输入的别名设置。

35. 完成更改后，在“文件”（File）菜单中选择“保存”（Save）。请注意，现在使用的是写字板。

36. 关闭写字板。

37. 在 RSLogix5000 的梯形图编辑器中，右键单击梯级 0 并单击“导入梯级”（Import Rung），如图 13-23 所示。

38. 将弹出“导入梯级”（Import Rung）窗口。选择文件 Z_A7_Conveyor_Export. L5X。然后单击“导入”（Import）。

39. 将弹出“导入组态”（Import Configuration）窗口，如图 13-24 所示。

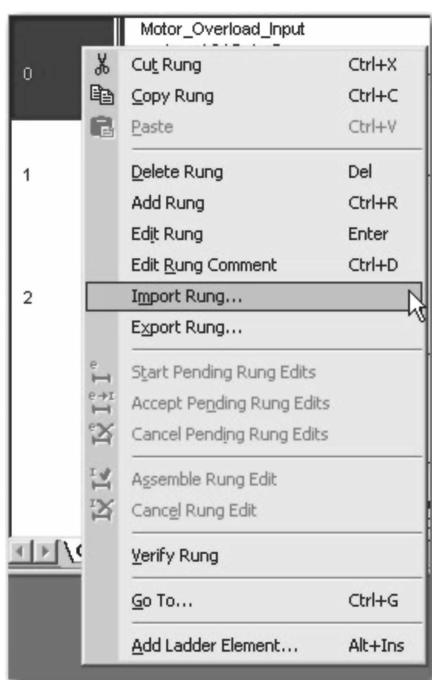
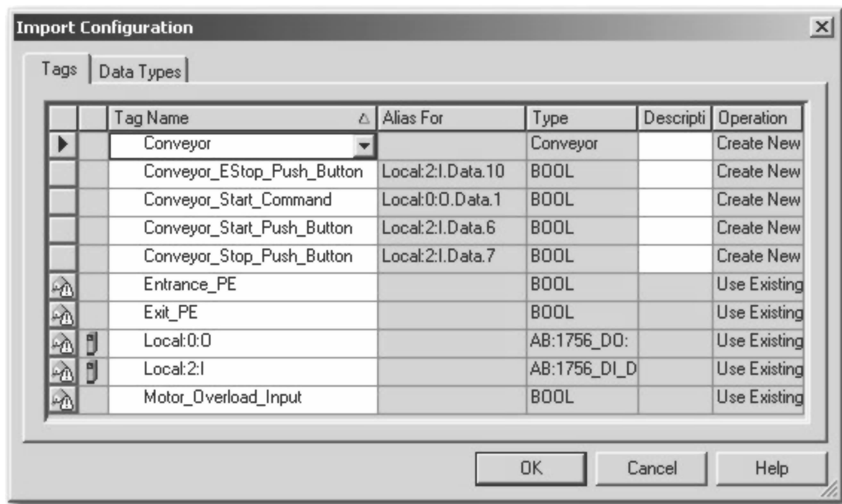


图 13-23 梯形图编辑菜单



Tag Name	Alias For	Type	Descripti	Operation
Conveyor		Conveyor		Create New
Conveyor_EStop_Push_Button	Local:2:I.Data.10	BOOL		Create New
Conveyor_Start_Command	Local:0:O.Data.1	BOOL		Create New
Conveyor_Start_Push_Button	Local:2:I.Data.6	BOOL		Create New
Conveyor_Stop_Push_Button	Local:2:I.Data.7	BOOL		Create New
Entrance_PE		BOOL		Use Existing
Exit_PE		BOOL		Use Existing
Local:0:O		AB:1756_DO:		Use Existing
Local:2:I		AB:1756_DI_D		Use Existing
Motor_Overload_Input		BOOL		Use Existing

图 13-24 “导入组态”窗口

40. 可看到“别名属于”（Alias For）列中的数据已更改，反映了之前在 L5X 文件中所做的编辑。为标签添加“3”，方法与之前的操作一样，如图 13-25 所示。

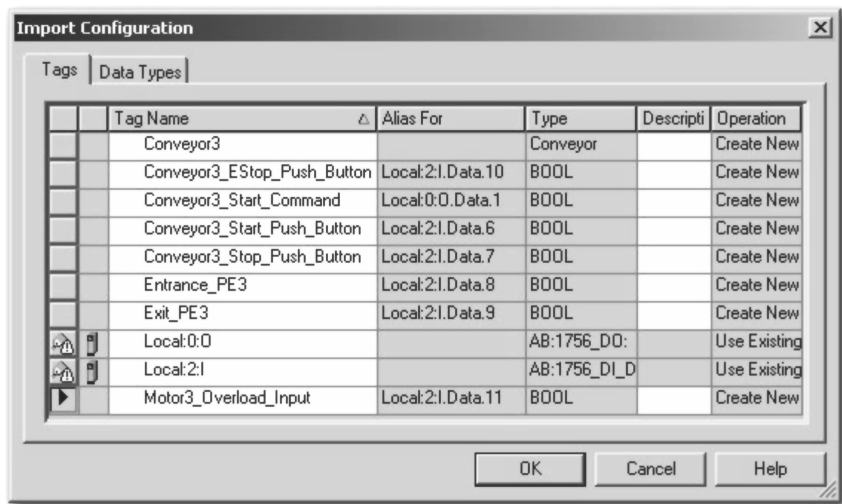


图 13-25 更改后的标签信息

41. 单击“确定”（OK）。

42. “导入”（Import）开始。完成后，便写好了 Conveyor3 的梯形图代码。所有标签都创建在合适的范围内；对于本例，该范围为 Conveyor。

43. 单击工具栏上的“校验控制器”（Verify Controller）图标，控制器验证后未发现错误。非常轻松地就添加了代码。

44. 在工具栏菜单中，选择“文件 > 退出”（File > Exit），不保存所做更改直接退出。

五、实验小结

通过使用部分导入/导出工具，几个小时就可以生成全部代码，而不再需要花几天时间。创建简单程序花费的时间不超过 25min。按照原有方式可能要花费数小时甚至几天时间。而且还要复制代码、更改地址、记录地址并验证所有操作的正确性。

有关 L5X 和 L5K 文件格式的更多信息，请参见出版物 1756-RM084-en-p. PDF。从中将了解到如何不使用 RSLogix5000 来创建 L5X 和 L5K 文件。

第二部分　SFC 编程

实验 14：创建 SFC 程序

一、实验目的

在本实验中，将执行以下操作：

1）构建一个简单的 4 步顺序功能图程序；

2）添加 SFC 元素（Step、Action 和 Transition）；

3）修改 Action 执行顺序；

4）向 SFC 例程添加注释；

5）测试 SFC 例程。

二、创建 SFC 例程

顺序功能图程序通常用作监管级控制算法。不同于执行完成时间在毫秒级的梯形图逻辑代码，SFC 中执行的 Step 可能需要几分钟、几小时甚至几天才能完成。

1. 在桌面上，打开 RSLogix 5000 

。

2. 使用图 2-1 中信息创建新项目，如图 14-1 所示。

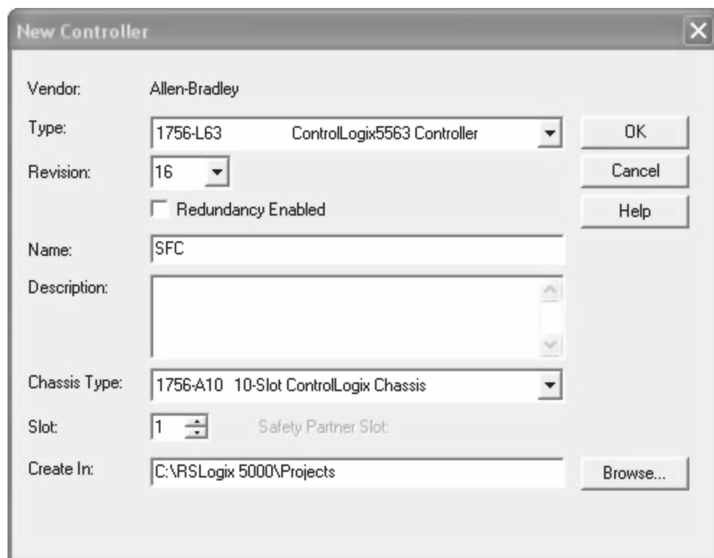


图 14-1　新建项目窗口

3. 然后，在 I/O 组态树中向插槽 2 添加 1756-IB16D 模块，向插槽 0 添加 1756-OB16D 模块。

现在已准备好创建 SFC 例程。

4. 右键单击“主程序”（MainProgram），然后选择“新建例程”（New Routine），如图 14-2 所示。

5. 按照以下内容将该项填写完整。确认已选择“顺序功能图”（Sequential Function Chart）作为例程类型，然后单击“确定”（OK），如图 14-3 所示。

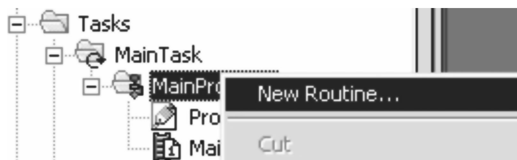


图 14-2 新建例程菜单

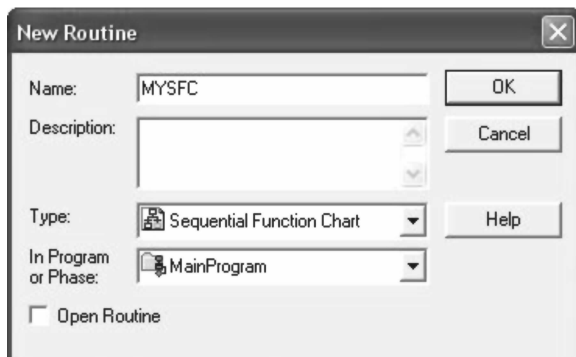


图 14-3 新建例程对话框

6. 在控制器项目管理器中，双击刚创建的例程将其打开。请注意，开始时即存在一个 Step 和一个 Transition。

1）Step 包含一系列想要执行的 Action；

2）Transition 是在执行下一 Step 之前，希望满足的条件。

举一个非常简单的例子，比如起动三个电动机的 Action，那么 Transition 可能是检查相应的模拟量输入，等电动机达到全速后再执行下一 Step，如图 14-4 所示。

注：第一个 Step 上套有两层方框。后面任何新插入的 Step 都只有一层方框。这个双层方框用来表示 SFC 从何处开始。

7. 右键单击 Step，然后选择“编辑“Step_000”属性”（Edit“Step_000”Properties），如图 14-5 所示。

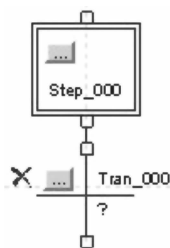


图 14-4 SFC 中的 step

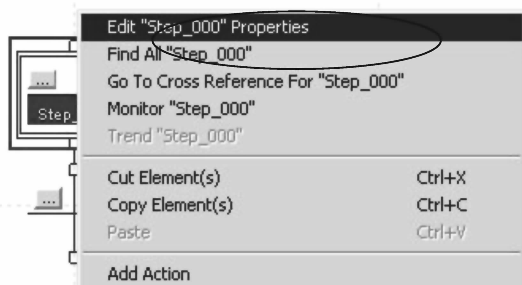


图 14-5 “Step”右键菜单

8. 请注意，这是程序范围内的标签，数据类型为 SFC_STEP。按“取消”（Cancel）以关闭此窗口，如图 14-6 所示。

9. 再次右键单击 Step，这次选择“添加操作”（Add Action）。

决定要编写一个简单的小例程来实现以下功能：

1）Step 0：打开 2 号灯和 3 号灯（在 ControlLogix 演示箱中）。

2）Transition 0：等待开关 13 接通。

3）Step 1：关闭 2 号灯和 3 号灯。

4）Transition 1：等待开关 14 接通。

5）Step 2：打开 4 号灯。

6）Transition 2：等待开关 15 接通。

7）Step 3：关闭 4 号灯。

8）Transition 3：始终为真，所以没有等待

刚刚添加了 Action，因此现在需要编写让 2 号灯和 3 号灯打开的代码。

10. 双击“操作”（Action）框中的问号，并按如图 14-7 所示输入逻辑。

注：在 Action 中输入的代码为结构化文本。可以键入下面显示的标签，也可以在框中单击右键并选择浏览标签。请记住，下面显示的标签均为控制器作用域。

1）Local:0:O. Data. 2: = 1;（所有行均以分号结尾）

2）Local:0:O. Data. 3: = 1;（这是赋值，因此必须是: = 而不是 = ）

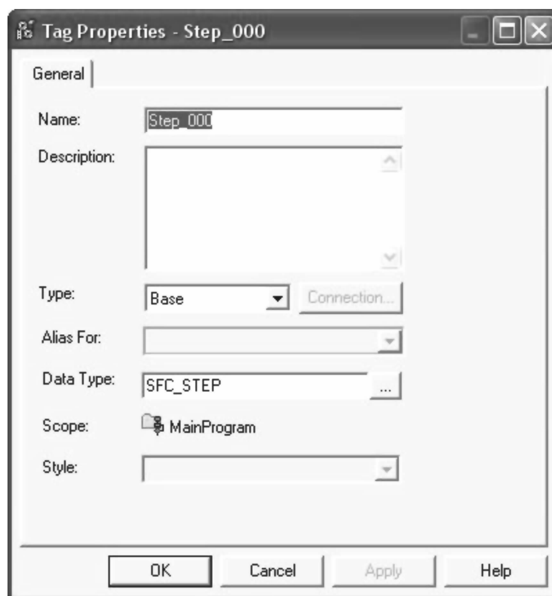


图 14-6 属性标签

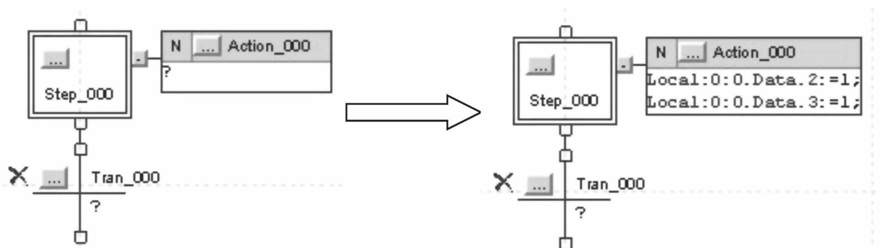


图 14-7 编辑“Action”

11. 现在双击 Tran_000 下方的问号，然后按如图 14-8 所示输入逻辑，Local:2:I. Data. 13（要等待开关 13 接通）。

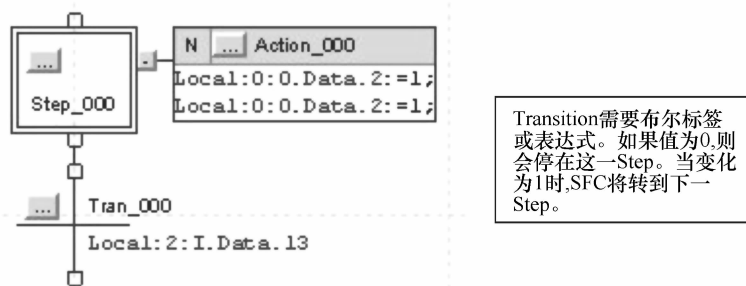


图 14-8 编辑“Transition”

12. 转到菜单栏并选择“文件→另存为→“SFCLABa””（File→Save As→“SFCLABa”），用新名称保存该项目。

13. 现在可以添加下一个 Step/Transition。按下“步骤 + 转换”（Step + Transition）图标。

14. 将它们拖动到初始 Step/Transition 的下方，注意如果距离足够近，则引脚会变为绿色。当引脚为绿色时松开鼠标，线会自动连接起来。

注：如果过早释放鼠标，两者未能连接在一起，只需先单击上引脚，再单击下引脚即可完成连接。

15. 拖拽新的 Step_001，注意线会随之一起被拖动，如图 14-9 所示。如果想要撤销拖拽并让它回到最初的位置，可以按 Ctrl + Z（撤销）。

请记住，想要完成：

1）Step 1：关闭 2 号灯和 3 号灯。

2）Transition 1：等待开关 14 接通。

16. 向 Step_001 添加两个 Action，按如图 14-10 所示填写完整。然后按照下方的建议隐藏/取消隐藏 Action。

1）Local:0:O. Data. 2: =0;

2）Local:0:O. Data. 3: =0;

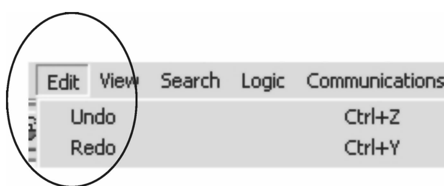


图 14-9 撤销命令

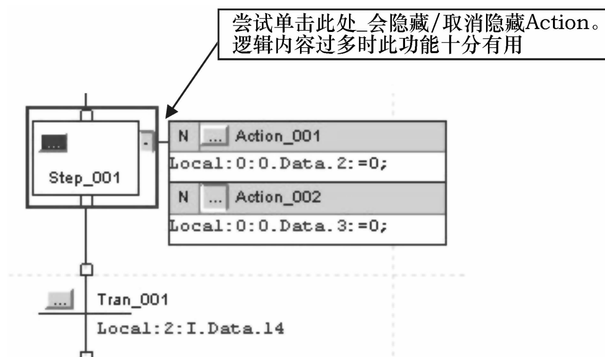


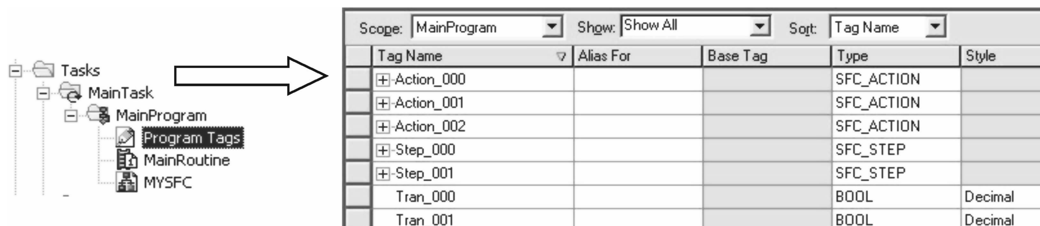
图 14-10 隐藏/取消隐藏 Action

注：在菜单栏中，可以选择“编辑”（Edit），然后选择显示或隐藏例程中的所有 Action。

17. 填写 Transition（见图 14-10）。

Local: 2: I. Data. 14

现在已经创建了一些逻辑，双击程序标签并查看通过逻辑创建的程序范围标签，如图 14-11 所示如前文所述，可以在程序所含的任意梯形图、功能块、SFC 或结构化文本例程中使用这些标签。



Tag Name	Alias For	Base Tag	Type	Style
Action_000			SFC_ACTION	
Action_001			SFC_ACTION	
Action_002			SFC_ACTION	
Step_000			SFC_STEP	
Step_001			SFC_STEP	
Tran_000			BOOL	Decimal
Tran_001			BOOL	Decimal

图 14-11 程序标签窗口

18. 在第一个 Step 中，在同一 Action 中完成了所有任务。在这一 Step 中，分两个 Action 完成。可以选择任意一种方式。甚至可以改变操作的顺序。回到逻辑，按下 Step_001 中带三个小

点的按钮。

19. 在“操作顺序”（Action Order）选项卡上，使用上/下箭头更改其顺序。

20. 按“确定”（OK），Action 的顺序已经发生变化。

21. 之前编写例程的时候，因为未添加注释而带来诸多不便。SFC 有多种方法可添加注释。现在具体了解一下。在菜单栏中，选择“编辑→添加 SFC 元素”（Edit→Add SFC Element）（Alt-Ins）。

22. 向下滚动并选择“文本框”（Text Box）。

23. 按“确定”（OK），并输入图 14-12 中内容。

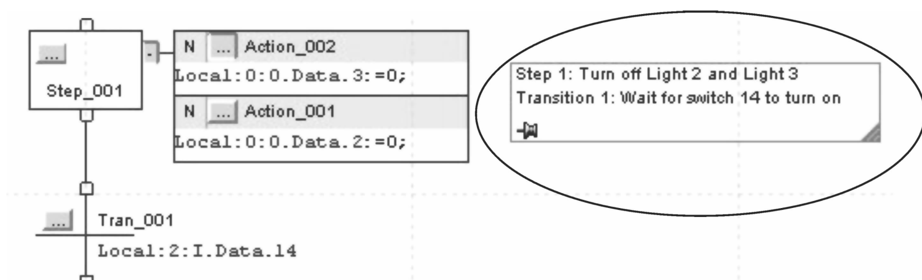


图 14-12　添加注释

刚刚创建了一个自由浮动的文本框。

24. 这似乎是一件很常见的事，因此觉得工具栏上应该有一个对应的图标，如图 14-13 所示。

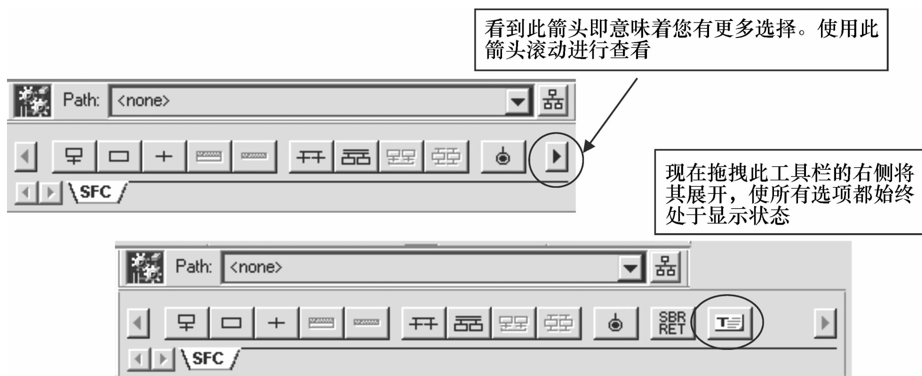


图 14-13　添加注释按钮

现在可以看到图标了！

25. 使用“文本框”（Text Box）图标为 Step_000，由编程人员自行添加注释。

26. 将鼠标悬停在小图钉图标的上方，注意显示的是已分离。

27. 单击图钉图标并拖拽着线在周围移动。

请注意，移动到 Step 上方时它会变为绿色。

28. 在 Step 内单击以固定注释。

29. 刚刚创建了一个固定的文本框。

30. 请注意，图钉图标已经改变，显示文本框已固定。

31. 尝试在附近移动该 Step，注意文本框会随之移动，如图 14-14 所示。

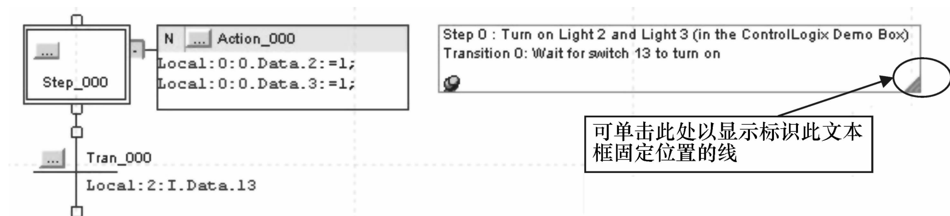


图 14-14 固定文本框

32. 将项目另存为“SFCLABb”。

33. 在菜单栏中，选择“编辑→复制元素”（Edit→Copy Element）。

34. 下一步，“编辑→粘贴”（Edit→Paste）。

35. 将 Step_002 拖拽到上一个 Transition 下方，当引脚变绿时释放。这会将它们自动连接起来。

36. 用相同的操作复制 Tran_000。

发现存在问题。现在 Step_002 外有两侧方框，而 Step_000 没有。复制了初始 Step，因此得到了另一个初始 Step。不能拥有两个初始 Step，所以 RSLogix 5000 不得不选择其中一个。稍后，等有机会查看错误消息时，才会解决此问题。

37. 暂时，请按照图 14-15 所示的截图修改逻辑：

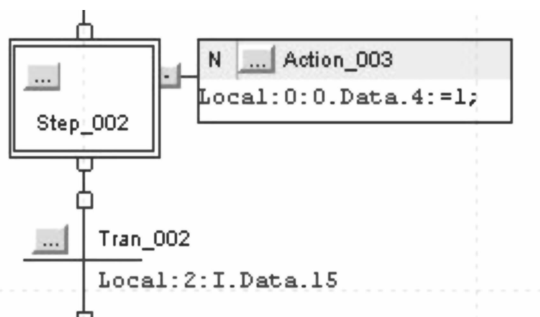


图 14-15 控制逻辑的修改

1） Step 2：打开 4 号灯（Local:0:O. Data. 4：=1；）

2） Transition 2：等待开关 15 接通（Local:2:I. Data. 15）

38. 可以开始最后一个 Step/Transition 对了。按下键盘上的 Insert 键，选择“步骤 + 转换→确定”（Step + Transition→OK），如图 14-16 所示。

39. 将其连接到 Tran_002 下方，然后按照图 14-17 所示的截图进行设置：

1） Step 3：关闭 4 号灯（Local:0:O. Data. 4：=0；）

2） Transition 3：始终为真，所以没有等待（1）

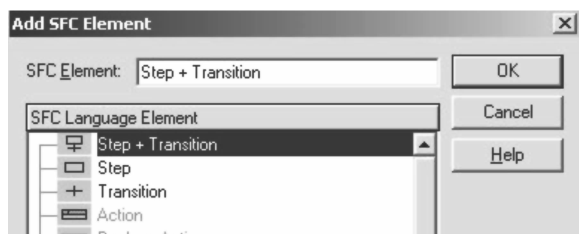


图 14-16 添加 SFC 元素对话框

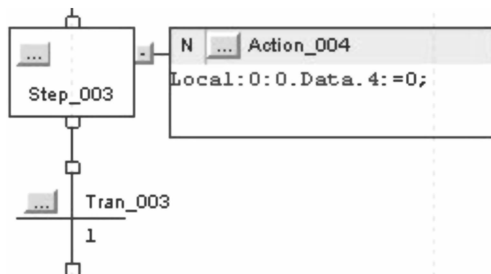


图 14-17 编辑“step”

注：如果未使所有 Step 和 Transition 彼此上下紧邻，则 Step_003 将超过（或者至少是接近）标识着分页的粗黑线。

虚线只是供参考的网格线。此编辑器的一大特色是仅通过滚动即可在页面之间移动。不必在

页面之间“跳转”。关于逻辑横跨页面的情况并没有任何限制，不必在页与页之间创建跨页连接器。不过，如果创建了跨页连接器，这样会减少很多连线，从而使编程界面看上去更加简明易读。

如果计算机连接有打印机并将逻辑打印出来，则第一页会是显示出所有逻辑的缩小视图。然后会打印出各个页面，并在每一侧配有小箭头以指明与之对应的页面。如果办公室略显单调，则可将所有页面贴在一起，在墙上构成一副壁画作为装饰。

事实上，唯一一点会限制 Step 数量的因素是控制器中可用内存的大小！这与某些人所熟悉的情况不同，因为 PLC-5 有固定的数量限制。

40. 将项目另存为“SFCLABc”。

41. 尝试用缩放图标同时查看更多逻辑。

42. 现在按下“检验例程”（Verify Routine）图标，注意到有三条错误，如图 14-18 所示。



图 14-18　错误信息

43. 第一个错误是因为不能以 Transition 作为逻辑的末尾。将通过添加停止来消除该错误。在指令工具栏中，按下停止图标并将其附加到逻辑的末尾，如图 14-19 所示。

下一个错误与之前已经注意到的问题有关。意外地使 Step_002 成为了初始 Step。这很容易更正：

44. 返回到 Step_000 并注意到它外面只有一层方框。

45. 单击带有三个小点的按钮得到属性页面。

46. 将类型更改为“初始”（Initial）并按下“确定”（OK）。

47. 请注意，现在它外面有两层方框，而同时 Step_002 被自动更正（RSLogix 5000 清楚只能有一个初始 Step），如图 14-20 所示。

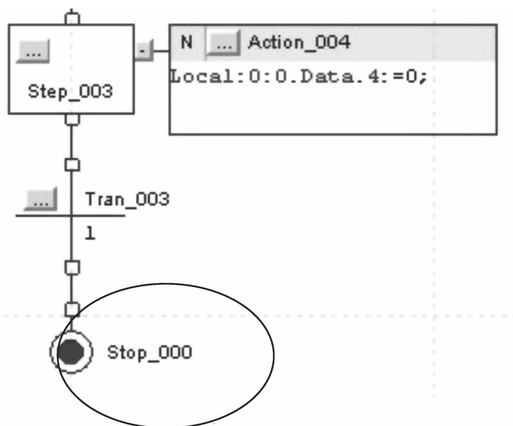


图 14-19　添加结尾

48. 现在按下“检验例程”（Verify Routine）图标，发现此例程中没有错误或警告（如果错误窗口未打开，请选择“视图→错误”（View→Errors））。

还有最后一件事要做。由于有多个例程，因此必须向该例程添加 JSR 指令。双击“主例程”（MainRoutine），如图 14-21 所示。

49. 按下键盘上的 Insert 键，键入“JSR”并按 Enter。

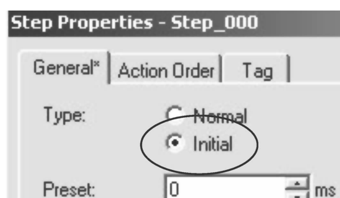


图 14-20 Step 属性对话框

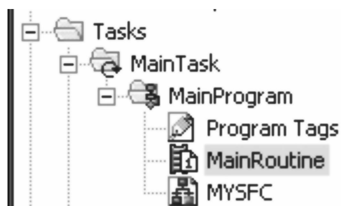


图 14-21 项目列表

50. 在文字 Routine Name 旁边→双击问号→从下拉列表中选择 MYSFC→按 Enter→单击 Return Par 旁边的问号，如图 14-22 所示。

51. 不想向此例程传送任何参数，因此不需要 Return Par 这行。删除该行的最简单的方法是单击 Return Par 旁边的问号并双击 Enter。对 Input Par 重复相同的操作，结果如图 14-23 所示。

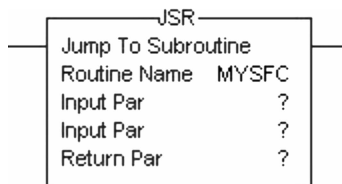


图 14-22 JSR 指令

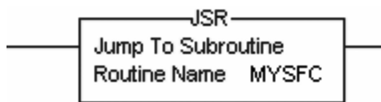


图 14-23 JSR 指令

52. 将项目另存为“SFCLABd”。

53. 将开关 12、13、14 和 15 打到左侧将其断开，然后下载到插槽 1 中的 L63 控制器。

54. 将控制器置于运行模式，并确认 2 号灯和 3 号灯已点亮。查看 SFC 例程并注意到 Step_000 已突出显示，表明逻辑执行到该 Step。这是因为 Tran_000 正等待接通开关 13。

Step 0：打开 2 号灯和 3 号灯（在 ControlLogix 演示箱中）

Transition 0：等待开关 13 接通

Step 1：关闭 2 号灯和 3 号灯

Transition 1：等待开关 14 接通

Step 2：打开 4 号灯

Transition 2：等待开关 15 接通

Step 3：关闭 4 号灯

Transition 3：始终为真，所以没有等待

55. 在观察逻辑的同时，接通开关 13。注意到灯熄灭了，现在 Step_ 001 被突出显示。

56. 接通开关 14 并注意到 4 号灯亮起。

57. 接通开关 15 并注意到 4 号灯熄灭，由于此 Transition 始终为真，因此 SFC 已结束。现在“停止”已突出显示。

如果这是 PLC-5，则逻辑会从顶部自动重新开始。在 ControlLogix 中，如果需要，必须手动绘制一条线连回到第一个 Step。

实验 15：SFC 执行选项

一、实验目的

在这部分实验中，将执行以下操作：

1）修改 SFC 执行；

2）使用结构化文本例程；

3）使用 Action Qualifier 和命名规则。

请按以下步骤完成实验 16。

二、修改 SFC 执行

使用结构化文本（STX，一种基于 Pascal 的编程语言）来深入了解一下 SFC 的执行方式以及可选的 Action Qualifier。

1. 将控制器更改为远程编程模式，然后离线。

2. 按下“控制器属性”（Controller Properties）图标并单击“SFC 执行”（SFC Execution）选项卡。在这里确定所期望的 SFC 执行方式：

1）仅执行当前激活 Step；

2）执行直至 False Transition（出现）。

如果选择“执行直至 False 转换（出现）”（Execute until False transition（is reached）），请务必牢记，SFC 开始后，在遇到 False Transition 或停止（或者任务看门狗定时器超时）之前，SFC 将一直执行下去。

3. 如图 15-1 所示，另一个选项是“仅执行当前激活步骤”（Execute current active steps only）。

4. 按下“确定”（OK），关闭此窗口并返回到 SFC 例程。

5. 按下“例程概览”（Routine Overview）按钮。

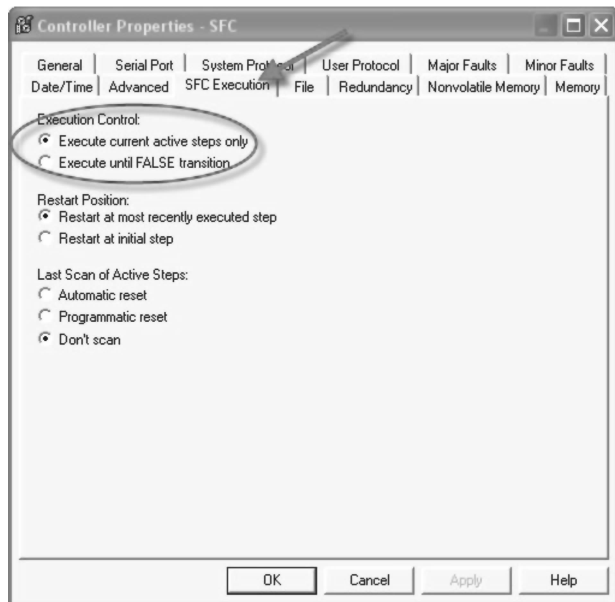


图 15-1　SFC 执行选项卡

请注意，可以得到例程的鸟瞰视图，所在的部分有相应的标记。可以在此视图中单击以快速浏览 SFC 的其他部分。还可以调整矩形的大小来进行缩放。

6. 关闭该视图，选择 Step_003 指令并将其向一侧拖拽几英寸。

7. 右键单击引线，将 Step_003 连接到上方的 Transition。

8. 选择“隐藏线”（Hide Wire）。

请注意为自动生成的热链接，它会提供坐标以便让找到另一端。

9. 在任意空白位置单击，热链接不再处于选中状态。

10. 将鼠标悬停在热链接上方，直到光标变为手形。单击左键，会跳转到另一端。这是一种在逻辑中进行跳转的好方法。

11. 右键单击热链接并选择“显示线”（Show Wire）来恢复显示。

已经完成了 SFC 的简单测试。可进一步了解一下结构化文本，结构化文本可以不嵌入到 SFC 中。结构化文本可用作独立的例程，正如梯形图逻辑或功能块例程一样。会创建一个结构化文本例程，不过要首先先来创建几个在例程中会用到的标签。

12. 双击“Program Tags”文件夹以打开标签数据库，并按如图 15-2 所示添加两个别名标签。

	conveyor_on	Local:2:I.Data.2(C)	Local:2:I.Data.2(C)	BOOL	Decimal
	light0	Local:0:O.Data.0(C)	Local:0:O.Data.0(C)	BOOL	Decimal
*					

Monitor Tags / Edit Tags

图 15-2 标签数据库

13. 使用以下信息创建一个新的结构化文本例程（提示-在“主程序”（MainProgram）上单击右键），如图 15-3 所示。

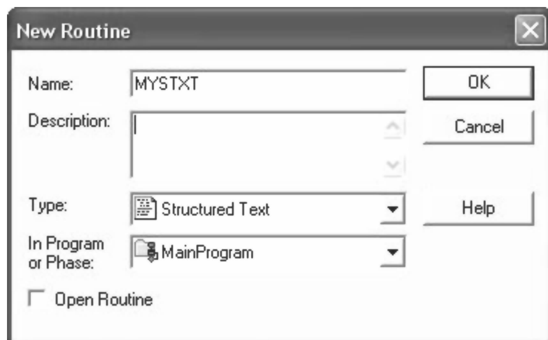


图 15-3 新建例程对话框

14. 双击打开刚创建的例程。会看到可输入结构化文本代码的空白页。

15. 按照内容输入结构化文本代码，如图 15-4 所示。

16. 检验例程以确保不存在编程错误。

现在已经输入完代码，所要做的只剩下进行测试。但在测试之前，必须明确如何调用并执行此代码。可以在主例程中输入 JSR 指令，像之前调用 SFC 例程时所做的那样。不过在本例中，将通过 SFC 调用此例程。不需要将希望执行的所有代码都直接输入到 SFC 中。相反，可以输入跳转指令从 SFC 内部跳转到其他例程。在 SFC 内部使用跳转而非直接在其中编写逻辑会有许多好处。

```
IF conveyor_on THEN light0 := 0;
  ELSE light0 := 1;
END_IF;
```

图 15-4 编辑结构化文本

可以从 SFC 中将逻辑分离出来。如果需要更改部分代码，不必编辑 SFC 本身，只需编辑包含代码的例程即可。

可以使用结构化文本之外的其他语言编写代码。由于结构化文本是唯一一种可直接嵌入 SFC 中的语言，因此如果希望使用梯形图逻辑或功能块来为部分过程编程，则必须使用跳转指令跳转到其他例程。

17. 打开“MYSFC”例程。

18. 向 Step 002 添加另一个 Action。

19. 在此 Action 中，按下鼠标右键并选择“设置 JSR”（Set JSR）。

20. 选择“MYSTXT”作为要跳转到的例程，然后选择“确定”（OK）。现在 Step 002 应当有两个关联 Action，如图 15-5 所示。

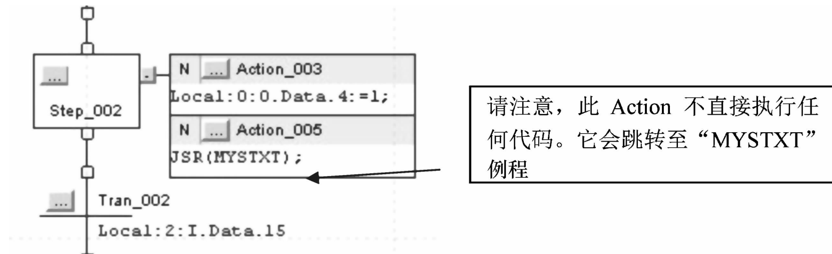


图 15-5 添加 Action

21. 将项目另存为“SFCLABe”并下载到控制器中。

现在可以测试项目了。

22. 将开关 12、13、14 和 15 打到左侧将其断开，然后下载到控制器。

23. 将控制器置于运行模式，并确认 2 号灯和 3 号灯已点亮。查看 SFC 例程并注意到 Step_000 已突出显示，表明逻辑执行到该 Step。这是因为 Tran_000 正等待接通开关 13。

Step 0：打开 2 号灯和 3 号灯（在 ControlLogix 演示箱中）

Transition 0：等待开关 13 接通

Step 1：关闭 2 号灯和 3 号灯

Transition 1：等待开关 14 接通

Step 2：打开 4 号灯并跳转到“MYSTXT”

Transition 2：等待开关 15 接通

Step 3：关闭 4 号灯

Transition 3：始终为真，所以没有等待

我们刚添加的新Action

24. 在观察 SFC 例程的同时，接通开关 13。注意到灯熄灭了，现在 Step_001 被突出显示。

25. 接通开关 14 并注意到 4 号灯亮起，如之前一样，但 0 号灯也同样亮起。

26. 观察例程“MYSTXT”，看能否判断出 0 号灯为何也会亮起（请记住，Step 2 中的新 Action 是跳转到例程“MYSTXT”）。

27. 如果现在按下演示箱上的按钮 DI2，认为会发生什么情况？（DI2 连接的是 Local:2:I. Data. 2）。如果不记得“conveyor_on”的别名，则可在结构化文本代码中移动鼠标并悬停在其上方，弹出的窗口中会显示其别名。

28. 按下演示箱上的按钮 DI2，确认 0 号灯熄灭。

29. 接通开关 15 并注意到 4 号灯熄灭，但 0 号灯是什么情况呢？认为它为何仍然是亮的？

实验结束。

三、额外任务

1. 再创建一个 Step/Transition 对。确认它们都在程序范围标签范围内，再删除 Step/Transition，对并确认标签也同样被删除。

2. 将部分 Step 和 Transition 重命名，使标签更易于理解和记忆。不必将它们保留为 Step_xxx。

3. 再深入一些。右键单击“MYSFC→属性→自动命名”（MYSFC→Properties→Auto Naming）。在这里可以为后续的 Step、Transition、Action 和停止设置默认名称，如图 15-6 所示。

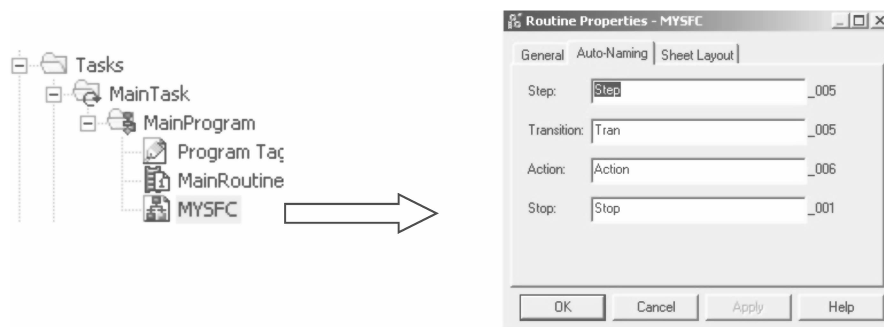


图 15-6　例程属性对话框

4. 额外添加逻辑复位 SFC（SFR 命令可以复位到任意 Step）。

5. SFP 命令可用于暂停 SFC 或取消暂停。

注：很多情况下，可能希望暂停（SFP）并复位到某一 Step（SFR），然后在准备好继续执行

时取消暂停（SFP）。如果只是想复位到某一 Step，则可不需要 SFP，使用 SFR 即可，如图 15-7 所示。

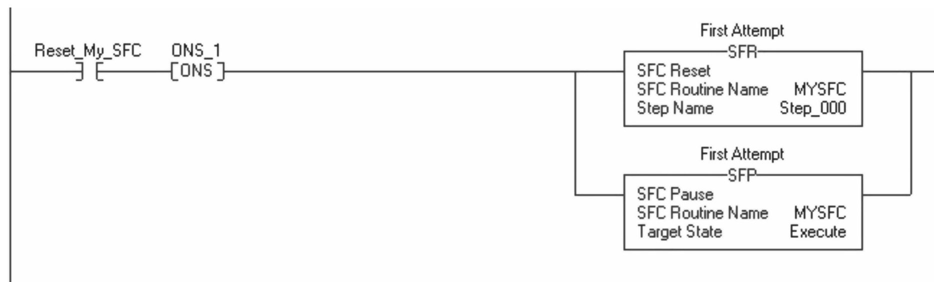


图 15-7 暂停 SFC

6. 分别在插槽 7 和 8 中添加模拟量输出和模拟量输入模块，然后通过再创建一个 Step/Transition 对并用以下值进行试验，来了解 Action Qualifier，如图 15-8 所示。

Local:7:O. Ch0Data: = Local:8:I. Ch0Data；延迟 10s

Local:7:O. Ch1Data: = Local:8:I. Ch1Data；无延迟

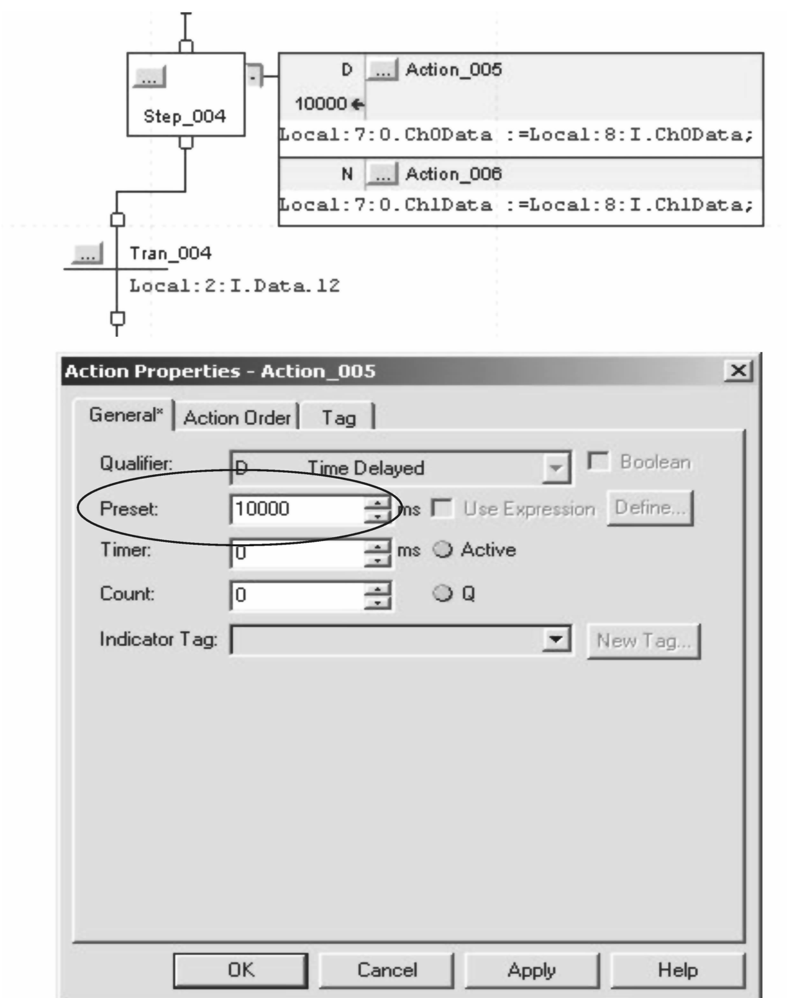


图 15-8 SFC 测试

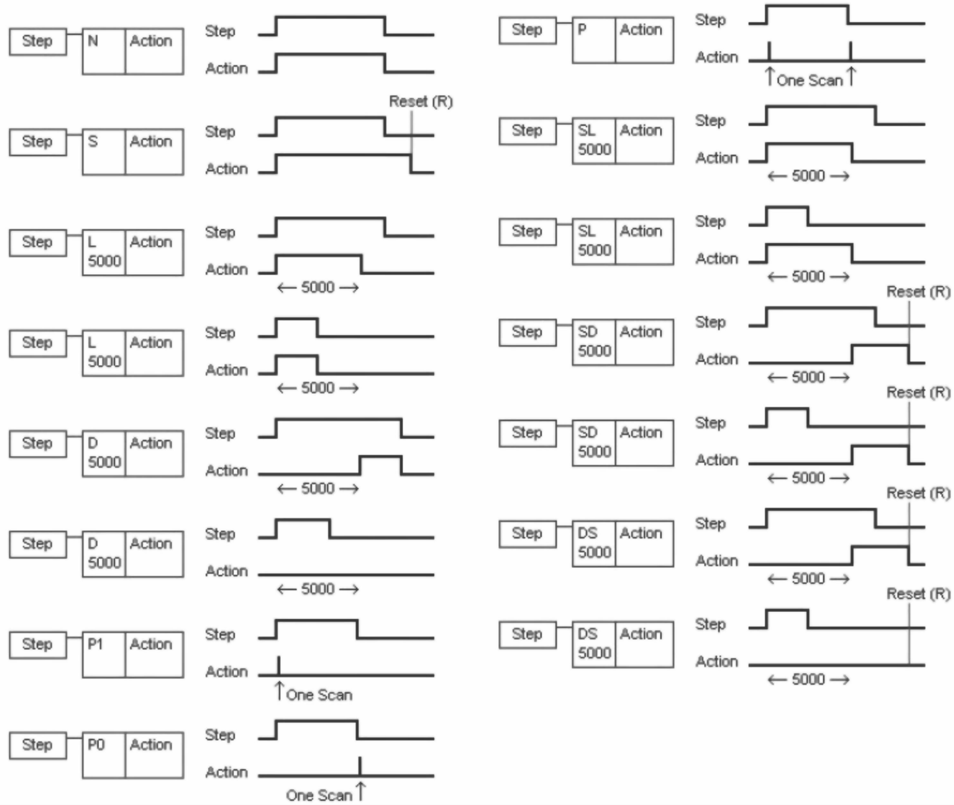

Action Qualifier Timing
Step
N
Action
Step
Action
Step
S
Action
Reset (R)
Step
Action
Step
L
5000
Action
Step
Action
← 5000 →
Step
L
5000
Action
Step
Action
← 5000 →
Step
D
5000
Action
Step
Action
← 5000 →
Step
D
5000
Action
Step
Action
← 5000 →
Step
P1
Action
Step
Action
One Scan
Step
P0
Action
Step
Action
One Scan
Step
P
Action
Step
Action
One Scan
Step
SL
5000
Action
Step
Action
← 5000 →
Step
SL
5000
Action
Step
Action
← 5000 →
Step
SD
5000
Action
Reset (R)
Step
Action
← 5000 →
Step
SD
5000
Action
Reset (R)
Step
Action
← 5000 →
Step
DS
5000
Action
Reset (R)
Step
Action
← 5000 →
Step
DS
5000
Action
Reset (R)
Step
Action
← 5000 →

图 15-8 SFC 测试（续）

第三部分　功能块编程

实验 16：功能块编程的简介

一、实验目的

本实验将演示功能块图的使用方法，该功能基于 IEC 61131-3 规范并已集成到 RSLogix 5000 软件中。在本实验中，将了解如何创建和编辑功能块图。

IEC 61131-3 和编程语言

IEC 61131-3 标准包括五种图形编程语言的规范。每种语言都能访问由其他语言编写的例程控制的数据。因此，功能块图可以直接访问由继电器梯形逻辑例程控制的计数器中的累加值。选择何种语言编辑器取决于具体应用。

继电器梯形图逻辑（RLL）

在其开发之初被用于取代现场的硬接线继电器和定时器。在外观上，它类似于电气示意图。任何使用继电器梯形图逻辑进行 PLC 编程的人都了解，RLL 的功能已经呈指数增长，如今包含不下 200 种不同的指令。Allen-Bradley 继电器梯形图逻辑应符合 IEC 61131-3 标准。

指令表（IL）

在审视过众多 PLC 制造商使用的低级语言（结构上类似机器汇编语言）之后，IEC 制定出了 IL 语言的规范。IL 的优势之一在于，可以使用这种语言的 PLC 在设计上非常简便。通常，无需中间软件编译器即可下载指令表程序。另一方面，IL 支持有限数量的程序流修改，不属于“高级语言”。在用户编程方面，继电器梯形图逻辑、功能块、结构化文本和顺序功能图均可转换为指令表，因此，罗克韦尔自动化不提供指令表编辑器。

功能块（FB）

功能块是控制系统的基本元素。FB 图外观类似于仪表与控制领域使用的 P&ID 图。功能块由两部分组成：一套输入、输出参数，以及特定类型功能块所特有的算法。功能块语言主要用于参数方面的控制，例如控制某一 PID 回路所需的所有输入/输出。IEC 61131-3 提出了一套标准功能块。与其他 PLC 制造商一样，罗克韦尔自动化在基本功能块基础上进行了扩展，从而能够针对各种类型的应用简便地进行配置。

顺序功能图（SFC）

与自身特点适合连续过程的功能块不同，顺序功能图语言以当前描述一系列事件的方法为基础。这种语言酷似最初用于计算机系统设计的一种图形系统，这种系统可以描述多阶段程序的行为。SFC 使用一系列 Step 和 Transition 来控制程序流。序列可以有合并或分支结构，并可与其他序列同时运行。虽然批次/配方控制可能是该语言的最主要应用，但其实任何由逻辑状态重复触发相似事件序列的场合都可以使用这种语言。

结构化文本（ST）

IEC 61131-3 结构化文本是一种高级语言，其语法与其他结构化文本语言（如 PASCAL）类似，非常适合管理各种各样的数据类型和行为。对于软件开发人员，ST 语言简单易学，编程和记录都十分方便。该语言的另一优势是能够编写复杂的数学算法。

二、启动 RSLogix 5000，创建一个新项目

在这部分实验中，将启动 RSLogix 5000 软件，并使用它创建一个新控制器文件。

重要说明：在开始之前，必须了解所使用的是哪个演示箱。

1. 在计算机桌面上，双击 RSLogix5000 图标启动 RSLogix5000 软件。显示 RSLogix5000 界面。

2. 在工具栏菜单中，选择“文件 > 新建”（File > New）。弹出“新建控制器”（New Controller）对话框。

3. 出现“新建控制器”（New Controller）界面后，请按以下步骤配置控制器：

1）从“类型”（Type）下拉菜单中选择学员站的控制器类型；

2）在“版本”（Revision）下拉菜单中指定 16；

3）在“名称”（Name）字段，键入“Language_Example”；

4）如果需要，请配置控制器的插槽号。

4. 单击“确定”（OK）接受所做更改，并退出“新建控制器”（New Controller）对话框。

现在，控制器项目管理器应如图 16-1 所示。

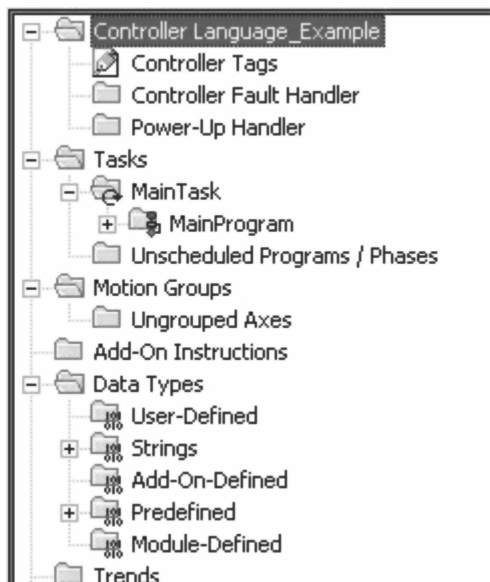


图 16-1　项目管理器结构

请注意，管理器类似于含文件夹结构的窗口型层级结构。

5. 在控制器项目管理器中，请花些时间找出控制器、任务以及 I/O 组态文件夹，因为随后将在本实验中引用这些内容。

三、创建和配置新的周期性任务和程序

在这部分实验中，将新建一个周期性任务并进行配置。

1. 在控制器项目管理器中，右键单击“任务”（Tasks）文件夹并选择“新建任务”（New Task）。弹出“新建任务”（New Task）对话框。

2. 按照图 16-2 所示输入参数，然后单击“确定”（OK）接受所做更改。

3. 在控制器项目管理器中，右键单击 FB_Task 并选择“新建程序”（New Program）。弹出“新建程序”（New Program）对话框。

4. 按照图 16-3 所示输入参数，然后单击“确定”（OK），接受所做的更改。

5. 控制器项目管理器中的“任务”（Tasks）文件夹应如图 16-4 所示（请注意，可能需要展开 FB_Task 文件夹）：

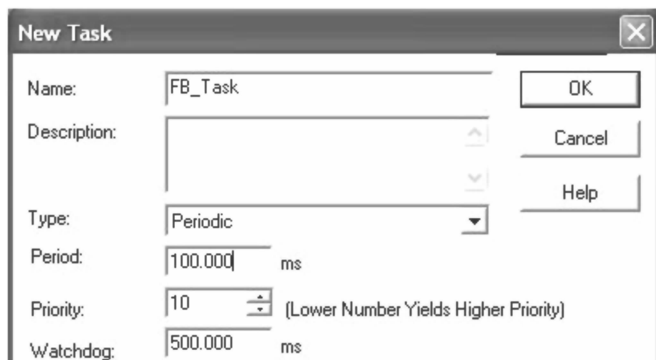


图 16-2　新建任务

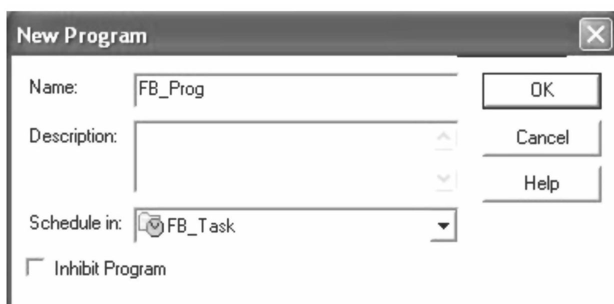


图 16-3 新建程序



图 16-4 项目管理器中的任务

四、创建功能块例程并安排其运行

例程是程序的基本容器。它可以由控制器支持的任何语言组成：梯形图、功能块、顺序功能图、结构化文本。很多情况下，这一阶段语言的选择往往取决于特定例程的实际功能，以及所支持语言在其特有的功能集方面是否适合完成这一任务。

1. 在控制器项目管理器中，右键单击 FB_Prog 并选择“新建例程”（New Routine）。弹出“新建例程”（New Routine）对话框。

2. 按照图 16-5 所示填写“新建例程”（New Routine）对话框，然后单击“确定”（OK）接受所做更改。

3. 现在，控制器项目管理器中的“任务”（Tasks）文件夹应如图 16-6 所示。

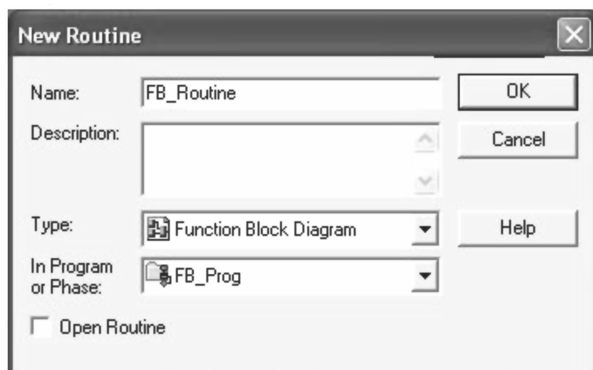


图 16-5 新建例程



图 16-6 任务文件夹

4. 要安排例程，请右键单击 FB_Prog 并选择“属性”（Properties）。将显示“程序属性”（Program Properties）对话框。

5. 单击“配置”（Configuration）选项卡，并在“主例程”（Main）下拉菜单中选择 FB_Routine 例程，如图 16-7 所示。

6. 单击“应用”（Apply），然后单击“确定”（OK），退出该对话框。

五、编辑功能块例程

1. 在控制器项目管理器中双击 FB_Routine 例程。

工作区中会打开一个空白工作表。

2. 在命名空间编辑框中将此表命名为“TIC101”，如图 16-8 所示。

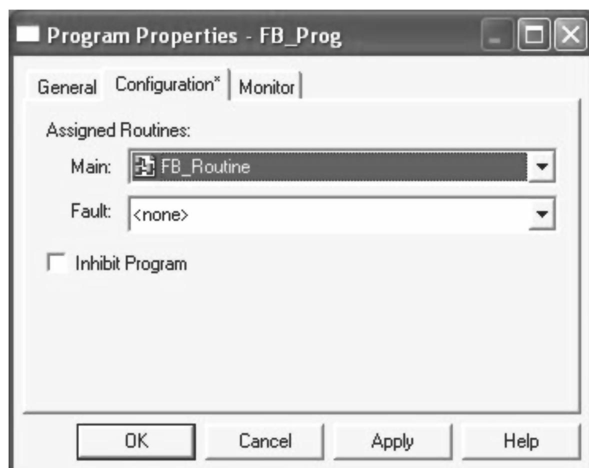


图 16-7　FB_Routine 例程



图 16-8　命名空间编辑框中

添加到图中的第一个块为增强型 PID 块（PIDE），用于控制模拟循环。

3. 在工具栏中的“过程”（Process）选项卡上，单击 PIDE 功能，如图 16-9 所示。

PIDE 块现在应显示在图中。



图 16-9　过程选项卡

4. 单击该块的“属性”（Properties）…按钮，花些时间查看一下所有可用参数。

注意：共分 7 个选项卡；“参数”（Parameters）选项卡列出了块中的所有参数。

第一列显示该参数是块的输入（I）还是输出（O）。而且在参数选项卡中，还可以使用第二列中的复选框来显示或隐藏功能块图中的参数引脚。

5. 单击“确定”（OK），关闭 PIDE 属性对话框。

6. 在工具栏上单击“输入线连接器”（Input Wire Connector），如图 16-10 所示。

7. 将输入线连接器（通过拖拽）移动到 PIDE 块的输入（左）侧并用线将其连接到 PV 点上，方法为单击一下输入参考输出引脚，再单击一下 PIDE PV 输入引脚。



图 16-10　输入连接器

注：可以单击并拖拽指令在表中任意移动。如果将鼠标悬停在有效连接点上方，则引脚会变为绿色。

8. 双击线连接器引用，打开输入标签名称的对话框，键入“PID_PV”并按回车接受，如图 16-11 所示。

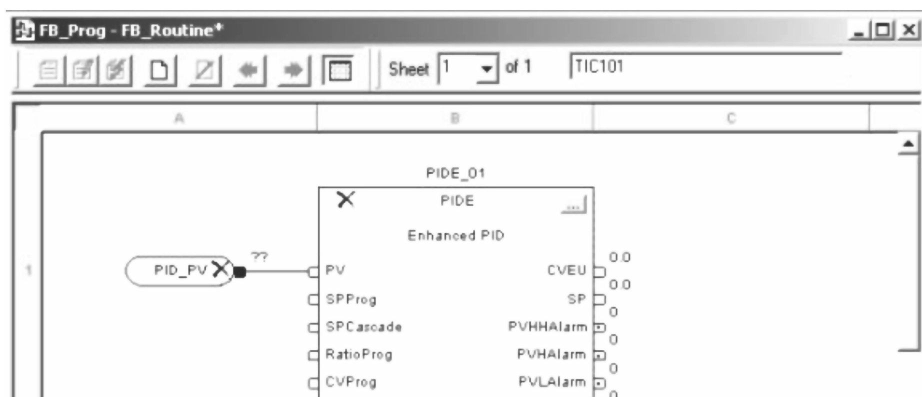


图 16-11 输入标签名称

9. 在工具栏上单击“输出线连接器”（Output Wire Connector），如图 16-12 所示。

10. 将输出线连接器（通过拖拽）移动到 PIDE 块的输出侧并将其连接到 CVEU 点上，方法为单击一下 PIDECVEU 引脚，再单击一下输出线连接器输入引脚。



图 16-12 输出连接器

11. 双击线连接器引用，键入“PID_CV”并按回车接受，如图 16-13 所示。

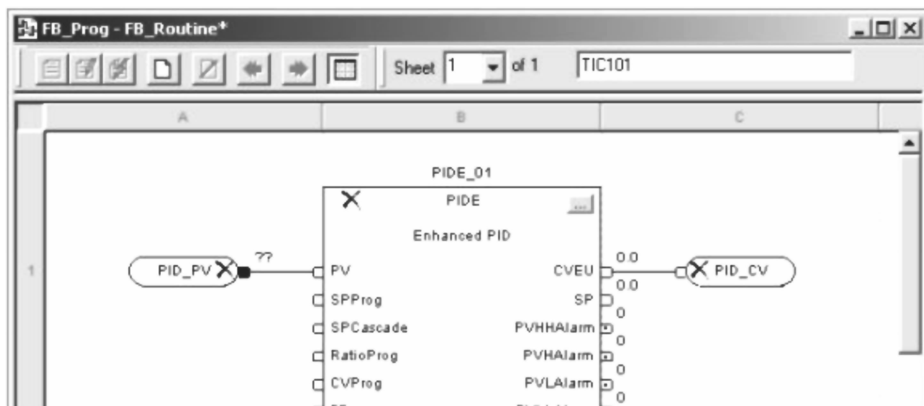


图 16-13 输出标签名称

12. 右键单击 PID_PV 输入线连接器到 PIDE_01. PV 的引线，并从弹出的列表中选择“假设数据可用”（Assume Data Available），如图 16-14 所示。

将这条线设置为“首先可用”数据点，意味着让表假定这条路径应首先评估。下一节中的模拟将创建到 PIDE_PV 参数的回路。

自整定功能需要关联 PIDE_AUTOTUNE 标签才能对回路进行自整定。

13. 在 PIDE 功能块的底部，双击？输入自整定标签名称。

14. 在字段中键入“PIDTune”，并按回车接受，如图 16-15 所示。

15. 右键单击 PIDTune 标签引用，选择“新建‘PIDTune’”（New “PIDTune”）创建自整定标签。

16. 按照图 16-16 所示填写“新建标签”（New Tag）对话框，然后单击“确定”（OK）接受所做更改。

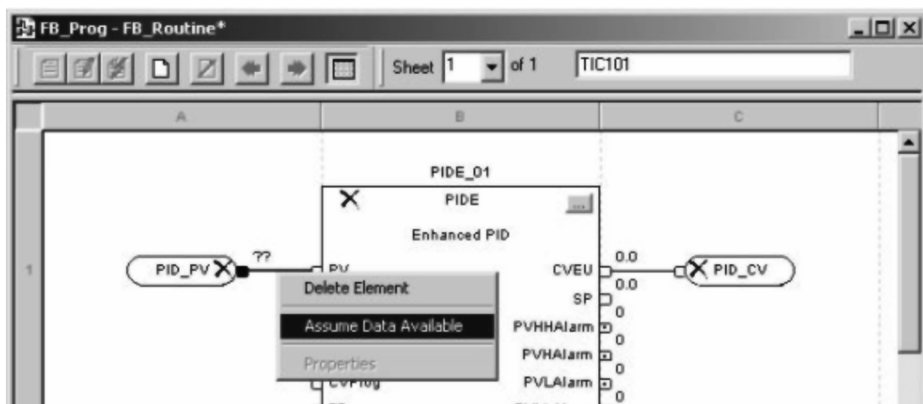


图 16-14　假设数据可用选项



图 16-15　键入“PIDTune”



图 16-16　“新建标签”对话框

17. 保存项目。

六、编辑功能块回路模拟例程

1. 单击“添加表”（Add Sheet）按钮，为模拟元素创建一个新表，如图 16-17 所示。

现在应当是一张空白表，指定表 2/2。此表中将包含模拟。

2. 将该表命名为“模拟”。

3. 在工具栏中的“过程”（Process）选项卡中，选择死区时间（DEDT）块并放置到表 2/2 上（可能需要在“过程”（Process）选项卡中使用滚动箭头查找 DEDT 指令）。



图 16-17　“添加表”按钮

接下来，将创建一个存储数组供该指令调用。

4. 双击 StorageArray 后面的?，键入“Dead_Array”并按回车接受。

5. 右键单击标签引用，选择“新建‘Dead_Array’”（New“Dead_Array”）创建死区时间数组标签。

6. 将此标签设置为 Real 型数组，在程序：FB_Prog 范围内，维数为 100，如图 16-18 所示。

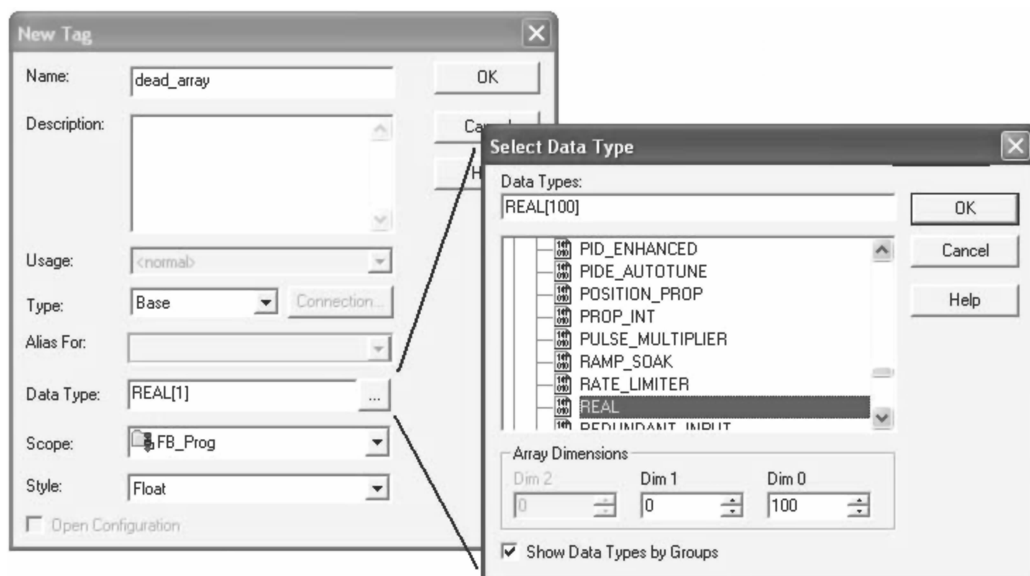


图 16-18　新建“Dead_Array”

7. 在两个对话框中单击“确定”（OK），接受所做更改。

8. 打开“死区时间属性”（DeadTime Properties）对话框（双击省略号），选择“参数”（Parameters）选项卡，将“死区时间”（Deadtime）设为 5.0 秒，将“增益”（Gain）设为 1.4，如图 16-19 所示。

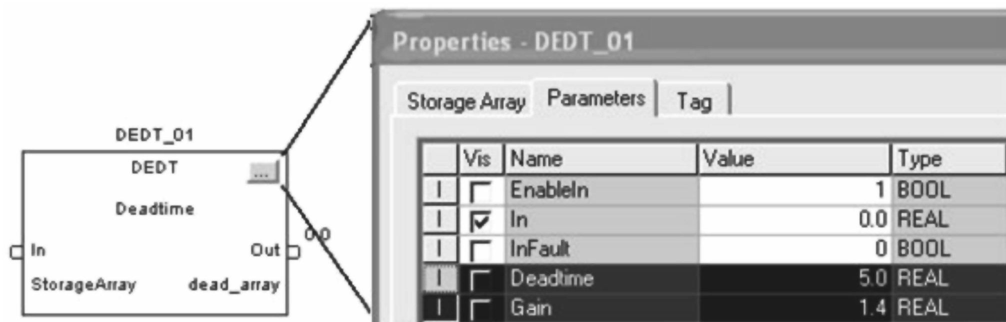


图 16-19　死区时间属性“对话框”

9. 应用更改并单击“确定”（OK）。

10. 在工具栏的“过程”（Process）选项卡上，选择超前滞后（LDLG）块并放置在表 2/2 中。

11. 打开 LeadLag 参数，双击省略号 …，将滞后配置为 40s，如图 16-20 所示。

12. 应用更改，并单击“确定”（OK）。

13. 将 DEDT_01. Out 连接到 LDLG_01. In，如图 16-21 所示。

14. 在工具栏中单击输入线连接器对象，并将其连接到死区时间块的输入上，如图 16-22 所示。

15. 双击线连接器引用，单击箭头弹出可用连接器引用的下拉列表，选择 PID_CV 并按回车接受。

LDLG Properties - LDLG_01

Parameters* | Tag

	Vis	Name	Value	Type	Description
I	☐	EnableIn	1	BOOL	Enable Input. If F
I	☑	In	0.0	REAL	The analog signal
I	☐	Initialize	0	BOOL	Request to initializ
I	☐	Lead	0.0	REAL	The lead time in se
I	☐	Lag	40.0	REAL	The lag time in sec
I	☐	Gain	1.0	REAL	The process gain
I	☐	Bias	0.0	REAL	The process offse

图 16-20 配置滞后时间

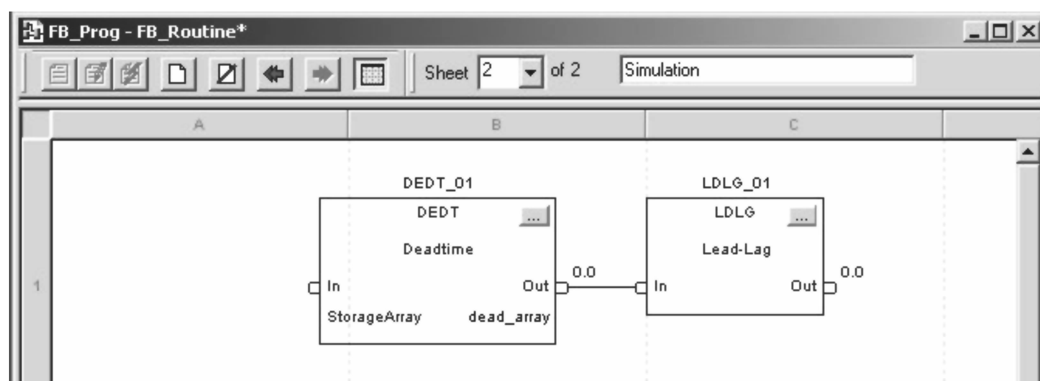


图 16-21 连接参数

16. 在表上拖放出一个输出线连接器，并将其连接到 LeadLag 的输出上，如图 16-23 所示。



图 16-22 输入线连接器



图 16-23 输出线连接器

17. 双击线连接器引用，单击箭头弹出可用连接器引用的下拉列表，选择 PID_PV 并按回车接受，如图 16-24 所示。

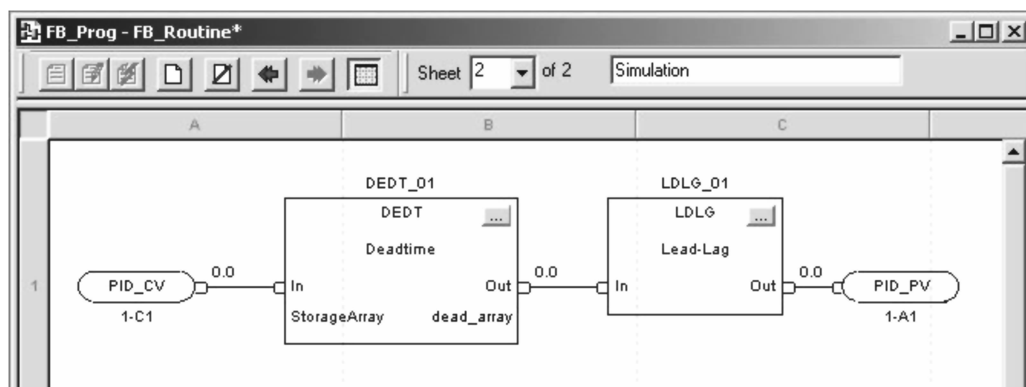


图 16-24 配置 PID_PV

18. 通过单击 RSLogix 5000 软件中的按钮来验证例程，如图 16-25 所示。

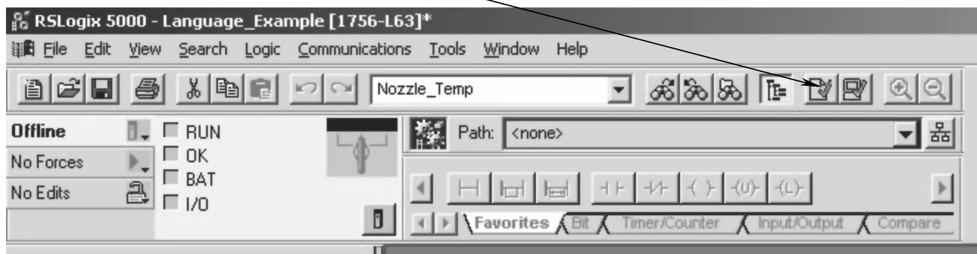


图 16-25 验证流程

19. 在工具栏菜单中，选择“文件 > 保存”（File > Save）。

20. 在工具栏菜单中，选择“通信 > 在线节点”（Communications > Who Active），然后浏览至学员站中的控制器。

21. 将程序下载到控制器，然后转到远程运行模式。会看到工作区上出现绿色的边框，表示控制器正在扫描此代码。

七、PIDE 回路自整定

1. 回到表 2/2，单击 PIDE 指令的属性按钮省略号，然后单击“整定”（Autotune）选项卡，如图 16-26 所示。

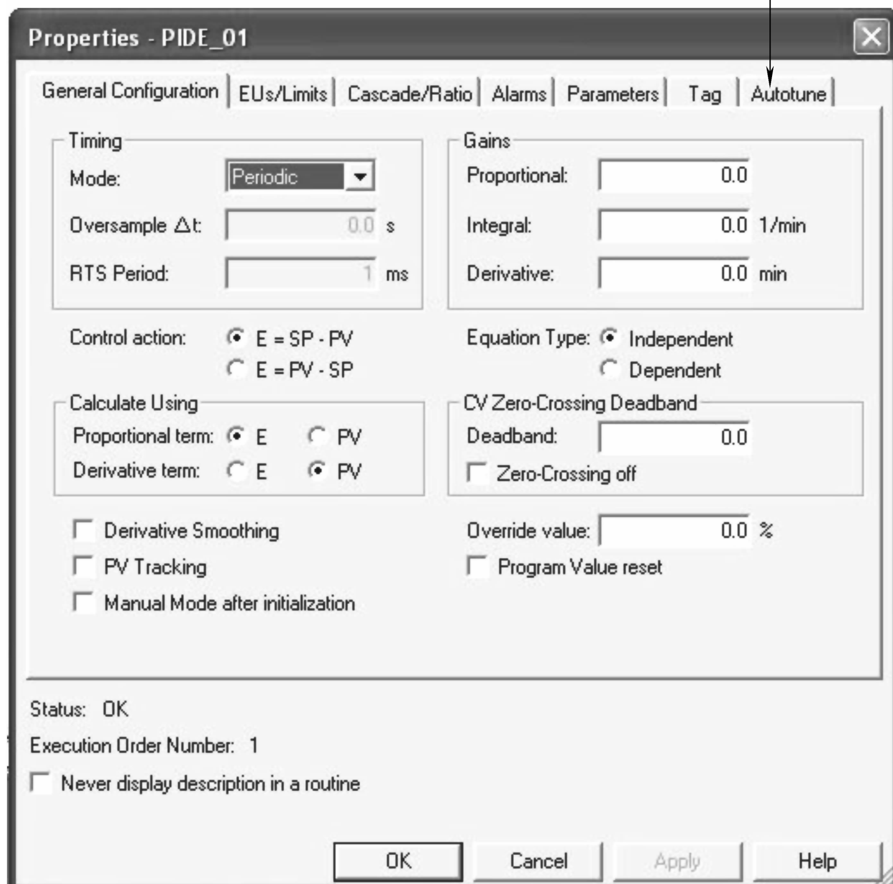


图 16-26 “整定”选项卡

2. 通过单击“获取标签”（Acquire Tag）按钮获取自整定资源（标签）。标签现在应当可用于此 PIDE 以及自整定，如图 16-27 所示。

参考信息

在 RSLogix 5000 中 PIDE 底部输入的自整定标签作为一项资源，既可以专属于单个 PIDE，也可以由许多 PIDE 共享。在共享资源时，对于某一次整定，同一时刻只有一个 PIDE 可以使用该自整定标签。如此一来，就必须有一种方法来“获取”要使用的资源，然后“释放”资源以便其他 PIDE 可以再次获取和使用。本段对话主要论述资源获取和随后的释放。要获取或释放资源，只需单击相应按钮。状态指示器显示自整定资源（标签）的当前状态。

3. 按照下图所示配置自整定，然后单击“应用”（Apply）接受所做更改，如图 16-28 所示。

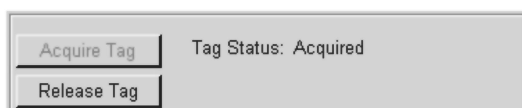


图 16-27　获取自整定资源

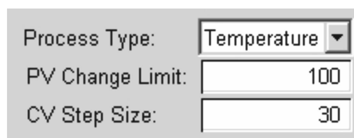


图 16-28　自整定配置

自整定通过将 PIDE CV 在其当前值基础上增加 30% 来调节“温度”过程，如果过程变量在自整定完成前会超过 100，则将中止自整定过程。

参考信息

“过程类型”（Process Type）指定要调节的系统为何种类型。这很重要，因为可能会影响选择用于自整定测试的模型类型（积分/无积分等）。

“PV 变化极限”（PV Change Limit）是针对所引用 PIDE 配置的绝对限值（以工程单位表示），如果系统在自整定过程中将超出该限值，则会停止（中止）自整定步骤。

“CV 步长”（CV Step Size）是自整定对 PIDE CV 进行更改的幅度，以便控制系统。在自整定执行期间，此处输入的数值（单位为百分比）将被累加到当前 CV 值上。

4. 点击“自整定”（Autotune）按钮，显示 PIDE 自整定启动界面，如图 16-29 所示。

5. 单击“启动”（Start）按钮，开始自整定过程，如图 16-30 所示。此过程可能耗时几分钟。

请注意，“执行状态”（Execution State）为“正在进行”（In Progress），“中止”（Abort）按钮是可用的。

6. 自整定完成后，首先应当确认“执行状态”（Execution State）为“完成”（Complete），并且各项增益应当如图 16-31 所示。

该部分显示基于上一次成功自整定的推荐增益，以及 PIDE 中使用的当前增益。

7. 通过选择慢对应的单选按钮，将“慢速响应”（Slow Response）这组增益加载到 PIDE 中。

8. 单击“设置 PIDE 中的增益”（Set Gains in PIDE）按钮，如图 16-32 所示。

请注意，PIDE 中的当前增益已经因为选择“慢速响应”而发生变化。

9. 关闭“自整定”（Autotune）和“整定”（Tune）对话框，以返回 FBD 的表 2/2。

请注意 PID_PV 值的变化。

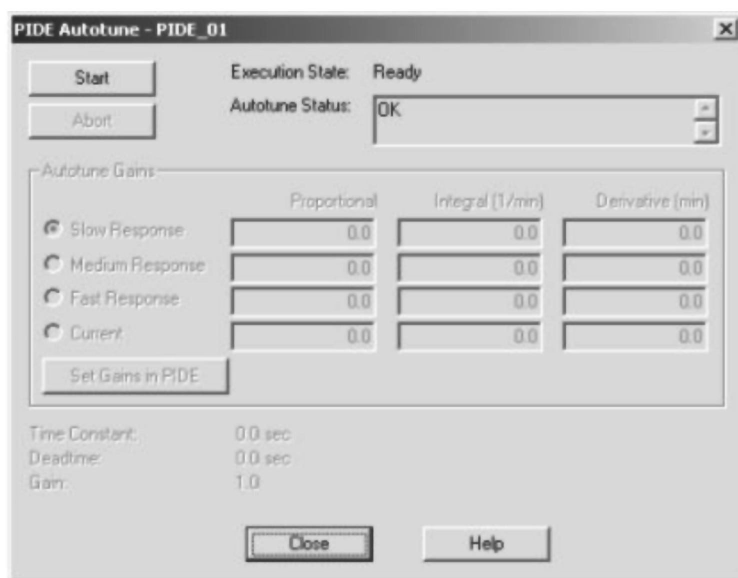


图 16-29 PIDE 自整定启动界面

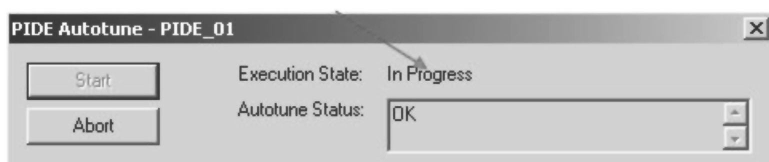


图 16-30 自整定过程

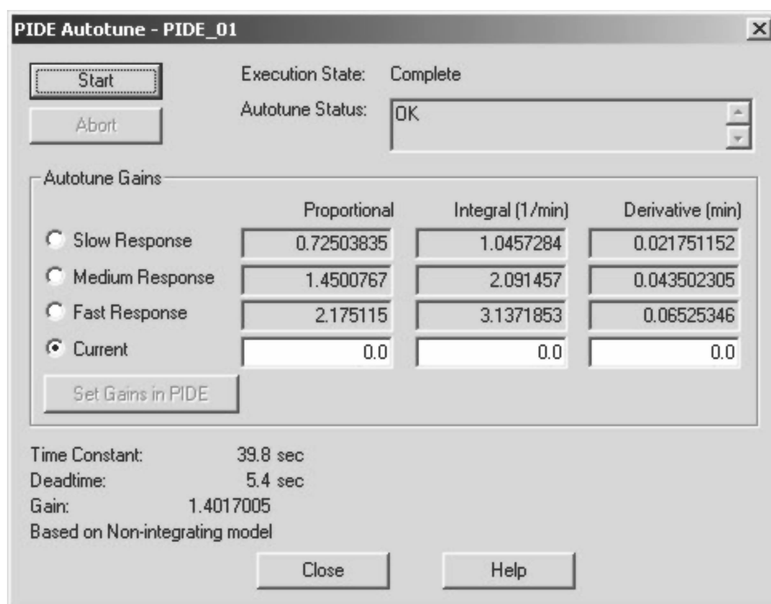


图 16-31 自整定完成

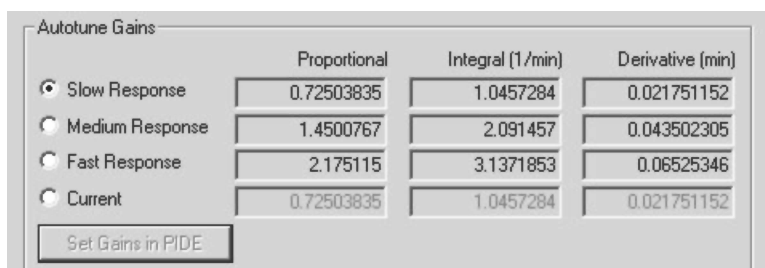


图 16-32　设置 PIDE 中的增益

八、创建趋势来跟踪过程变量

1. 在控制器项目管理器中，右键单击“趋势”（Trend）文件夹并选择“新建趋势”（New Trend）。

2. “新建趋势”（New Trend）窗口出现后，请按照图 16-33 所示输入趋势名称，然后单击“下一步”（Next）。

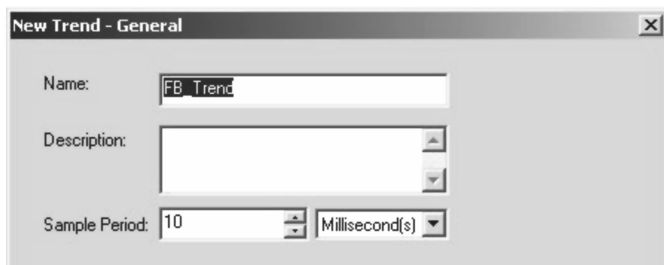


图 16-33　新建趋势

3. 出现如图 16-34 所示界面后，从“范围”（Scope）下拉菜单中选择 FB_Prog，然后展开“可用标签”（Available Tags）窗口中的 PIDE_01 标签。

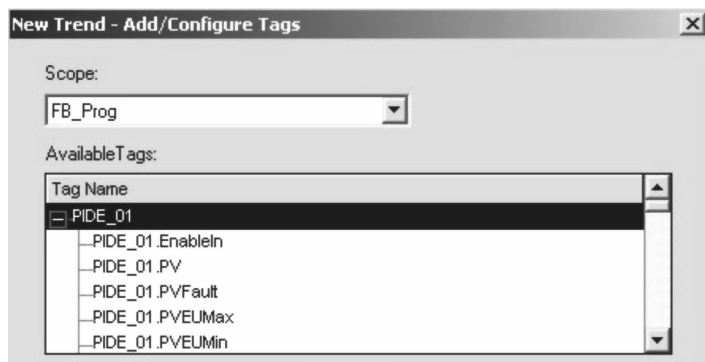


图 16-34　PIDE_01 标签

4. 从列表中选择 PIDE_01. PV 标签，然后单击“添加”（Add）按钮。
5. 对 PIDE_01. CV 和 PIDE_01. SP 元素重复步骤 4。配置窗口应如图 16-35 所示。
6. 单击“完成”（Finish）接受这些更改，出现的趋势如图 16-36 所示。
7. 右键单击黑色区域并选择“图属性”（Chart Properties）。

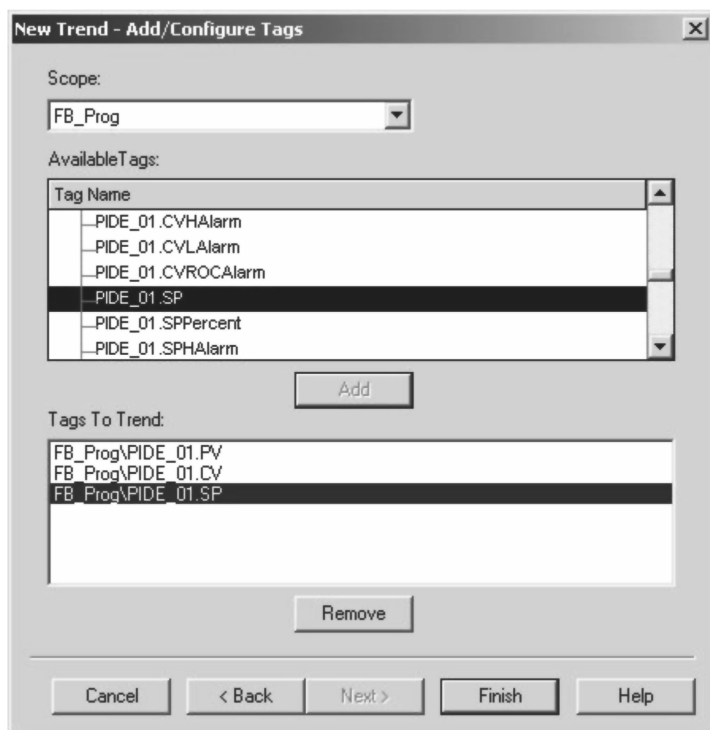


图 16-35 PIDE_01. CV 和 PIDE_01. SP 配置窗口

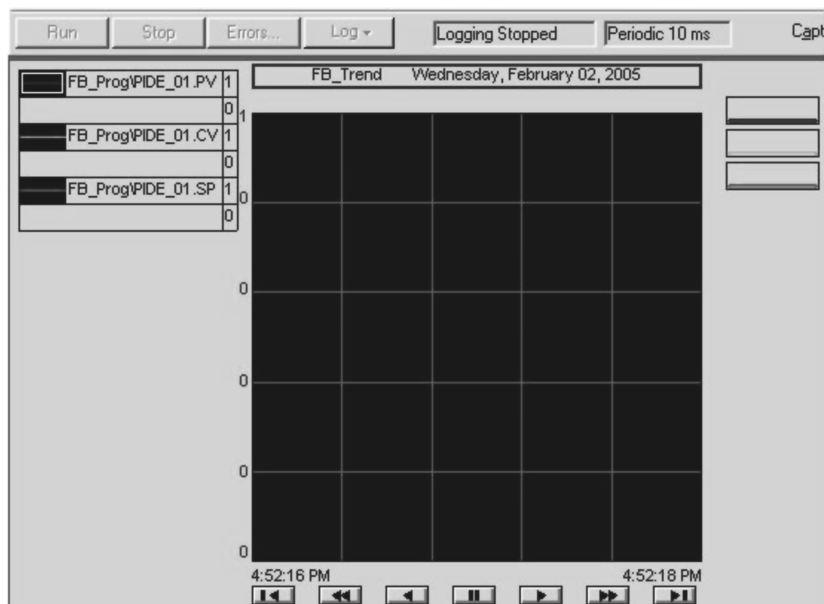


图 16-36 配置后趋势图

8. 单击“X 轴”（X-Axis）选项卡，将时间设为 3min，如图 16-37 所示。
9. 单击“Y 轴”（Y-Axis）选项卡。
10. 单击“自定义”（Custom）单选按钮，然后按照图 16-38 所示设置最大值和最小值。
11. 单击“应用”（Apply）接受所做更改，然后单击“确定”（OK）退出趋势属性窗口。

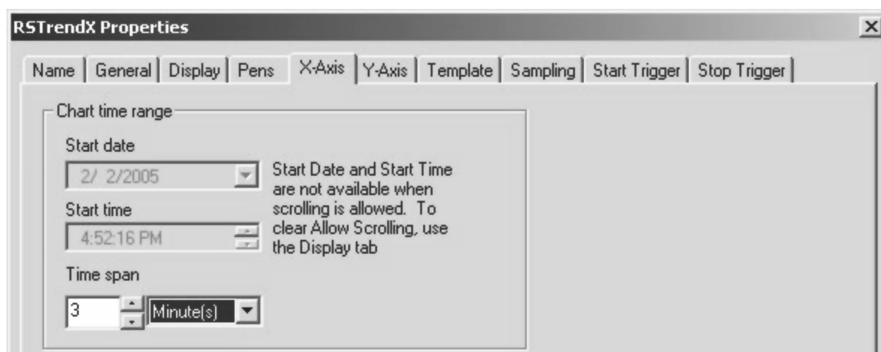


图 16-37　“X 轴”选项卡

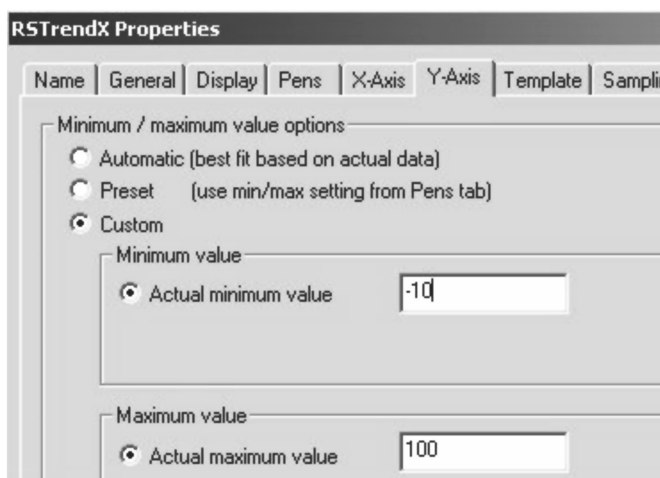


图 16-38　“Y 轴”选项卡

现在已完成趋势界面的定义；稍后将在本实验中用到。

九、功能块例程的在线编辑

在步骤方面，其在线编辑功能与梯形图例程的完全相同：

1）开始编辑所选例程（原始例程仍在运行，是在对副本进行编辑）；

2）根据需要对副本进行编辑；

3）验证（接受）副本中的编辑结果；

4）测试编辑后的副本（在执行线程中切换原始版本和编辑版本）；

5）可以选择取消测试返回执行原始版本；

6）对编辑的副本进行汇编以替换原始版本。

在功能块编辑器中，编辑“区域”为正在编辑的整个例程。因此，从编辑得到验证开始直到编辑被取消或经过汇编，所编辑的两个例程都会驻留在内存中。

1. 浏览至例程 FB_Routine，使指定表 2/2 出现在编辑器工作区中。

2. 在功能块工具栏中单击“开始待决例程编辑”（Start Pending Routine Edits）按钮，如图 16-39 所示。

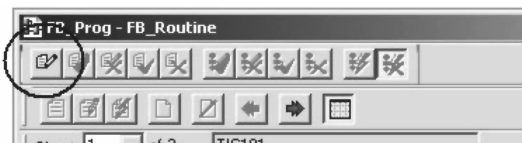


图 16-39　开始待决例程编辑

此时，应显示“待决编辑”视图如图 16-40 所示。

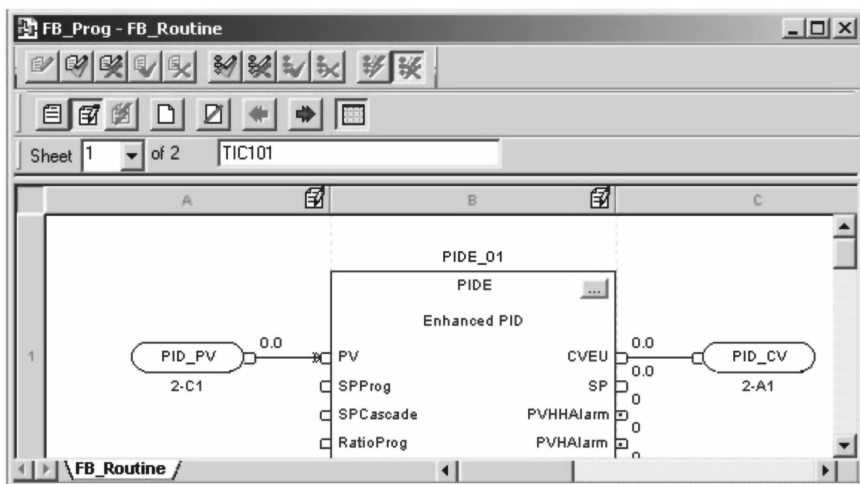


图 16-40 “待决编辑”视图

请注意，功能块图外围的框不再是绿色。这是因为程序未运行正在编辑的例程。仍在运行其原始版本。要查看原始版本，请单击“原始视图”（Original View）按钮，如图 16-41 所示。

要查看待决编辑视图，请单击“待决编辑视图”（Pending Edits View）按钮，如图 16-42 所示。

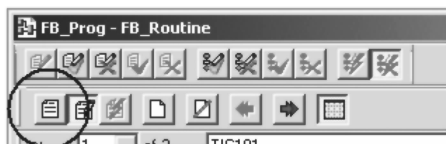


图 16-41 “原始视图”按钮



图 16-42 “待决编辑”按钮

3. 在指令工具栏中，将输入参考（IREF）放置在表中 PIDE 的左侧，并与 SPProg 输入相连，如图 16-43 所示。

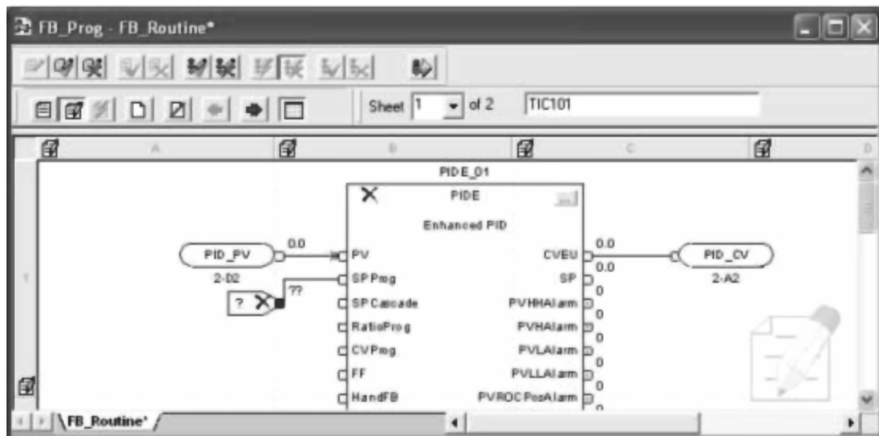


图 16-43 连接输入参考

4. 在 IREF 中键入 RemoteSP，将此标签创建为 REAL 类型，如图 16-44 所示。

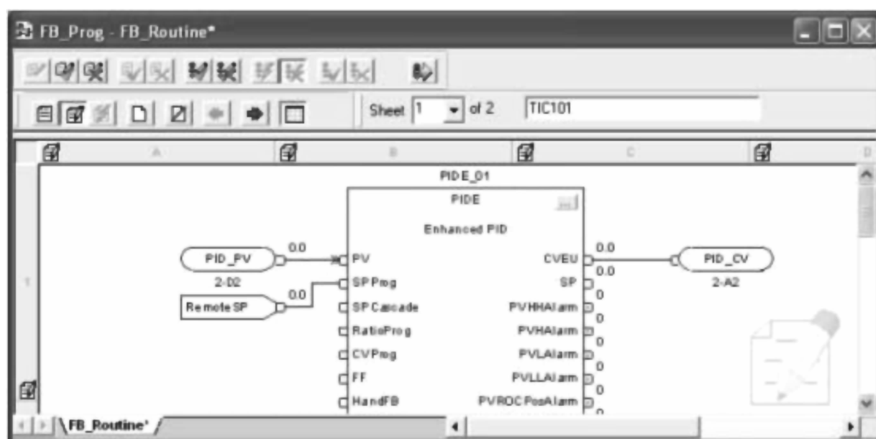


图 16-44　输入 RemoteSP

5. 单击“接受待决例程编辑”（Accept Pending Routine Edits）按钮，验证并接受所做编辑，如图 16-45 所示。



图 16-45　“接受待决例程编辑”按钮

现在，“编辑”（Edit）工具栏的可用选项将发生变化，显示例程已通过验证并可进行测试或将其取消，如图 16-46 所示。此外，视图会变为“测试?”视图，表外围的边框也会发生变化，表明可由测试激活。

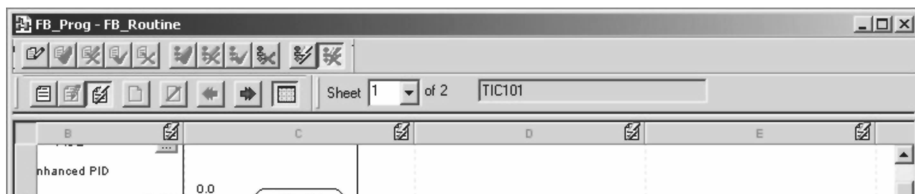


图 16-46　“测试”视图

注：如果“编辑”（Edit）工具栏不像上文所述，或者验证过程中出现错误，则请仔细检查本节中的步骤。

6. 单击“测试接受的程序编辑”（Test Accepted Program Edits）按钮，激活新例程来替换当前有效例程，如图 16-47 所示。

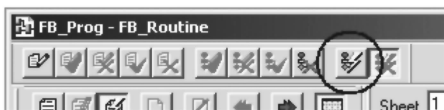


图 16-47　“测试接受的程序编辑”按钮

7. 弹出对话框询问是否测试程序编辑时，单击“是”（Yes）。请注意，所编辑表外围的边框已变为绿色，表示它现在是有效副本。

此时，可以浏览“原始”和“测试”视图，并可使用“测试”和“取消测试”功能。请勿取消编辑，否则将无法应用所做的编辑，除非将这一部分整个重复一遍。

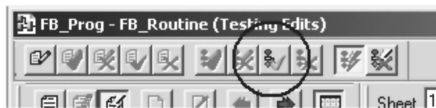


图 16-48　“汇编接受的程序编辑”按钮

8. 单击“汇编接受的程序编辑”（Assemble Accepted Program Edits）按钮将程序编辑应用到当前程序中并退出编辑模式，如图 16-48 所示。

9. 弹出对话框询问是否汇编程序编辑时，单击“是”（Yes）继续。请注意，所编辑表外围的边框上没有编辑符号了，这表示现在它在程序中是唯一的。

实验 17：通过 ActiveX 面板练习使用 PIDE

一、实验目的

RSLogix5000 提供七个 Active X 面板，可在 FTView 或任何 Active X 容器（例如 MS Excel）中使用。对于以下功能块均提供面板：Alarm、Enhanced Select、Totalizer、Ramp/Soak、Discrete 2 State、Device、Discrete 3 State Device 和 Enhanced PID。

二、在 RSLinx 中组态主题

1. 使用“向下还原”（Restore Down）按钮将 RSLogix 5000 项目最小化。

2. 在计算机桌面上，单击 RSLinx 图标。

3. 在 RSLinx 中的工具栏菜单上，选择“DDE/OPC > 主题组态”（DDE/OPC > Topic Configuration），如图 17-1 所示。



图 17-1 RSLinx 工具栏菜单

4. 出现以下窗口后，单击 Language_Example 主题，展开 AB_ETHIP_1 驱动程序，然后选择所在学员站的控制器，如图 17-2 所示。

注：如果不确定是哪个控制器，请询问你的指导老师。

5. 在“数据采集”（Data Collection）选项卡中，按照图 17-3 所示配置该主题。

6. 单击“应用”（Apply），然后在完成后单击“完成”（Done）（见图 17-3）。

7. 将 RSLinx 会话最小化。

现在，可以运行 Excel 并在工作表中放置 PIDE 面板了。

三、向 MS Excel 电子表格中添加 PIDE 面板

1. 在计算机桌面上，双击 Microsoft Office Excel 2003 图标。

将在屏幕上看到空白工作表。由于面板是 Active X 控件，因此必须在 Excel 中访问“控件工具箱”（Control Toolbox）才能向表中插入 Active X 对象。如果未显示“控件工具箱”，请转到

“视图”（View），选择“工具栏”（Toolbars），然后选择“控件工具箱”（Control Toolbox）。

2. 在电子表格的“控件工具箱”（Control Toolbox）中，选择工具箱底部的“更多控件”（More Controls）图标。

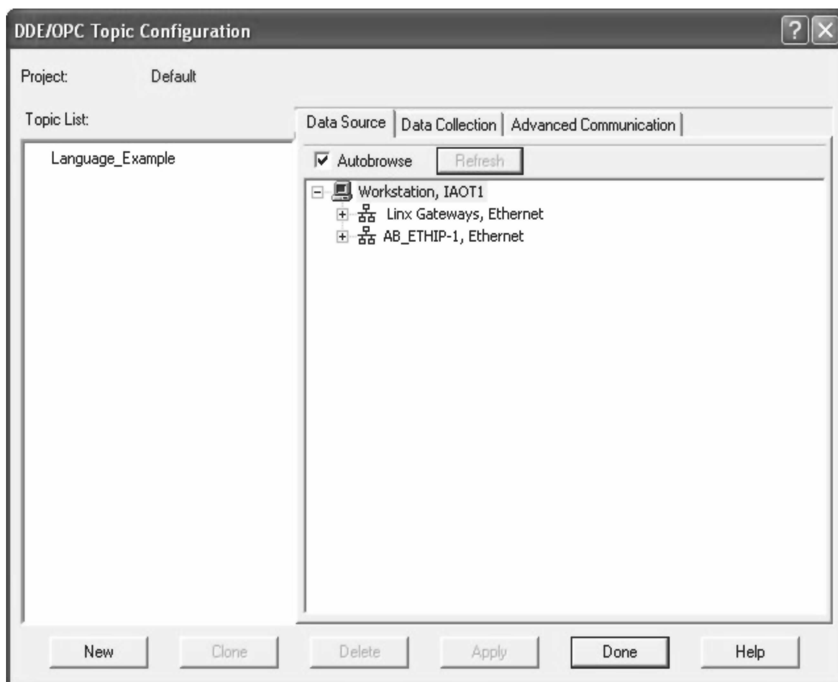


图 17-2　选择控制器

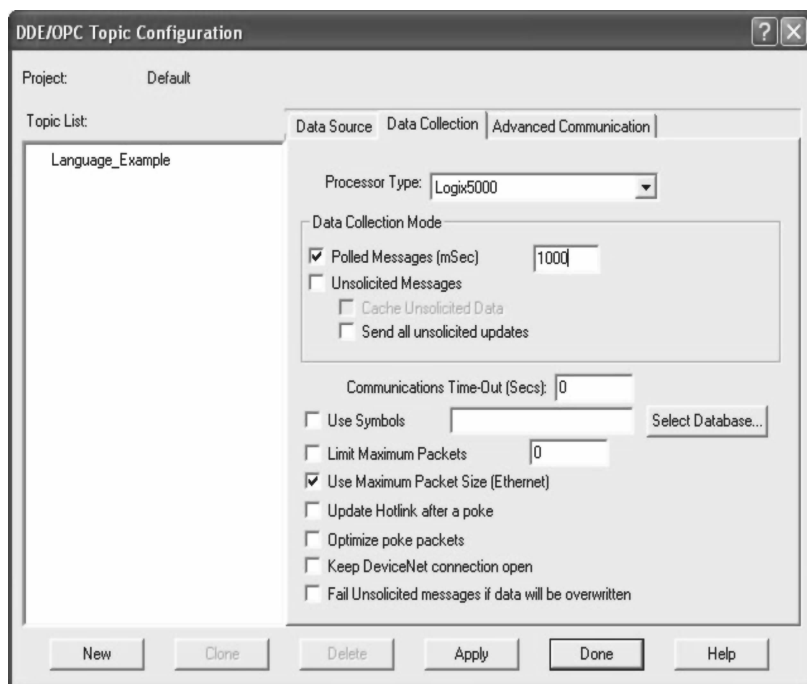


图 17-3　“数据采集”选项卡

3. 在弹出的滚动列表中，向下滚动并选择“Logix 5000 PIDE 面板控件”（Logix 5000 PIDE Faceplate Control）。光标现在应变为一个十字。

4. 在表中用光标画出一个方框，结果如图 17-4 所示。

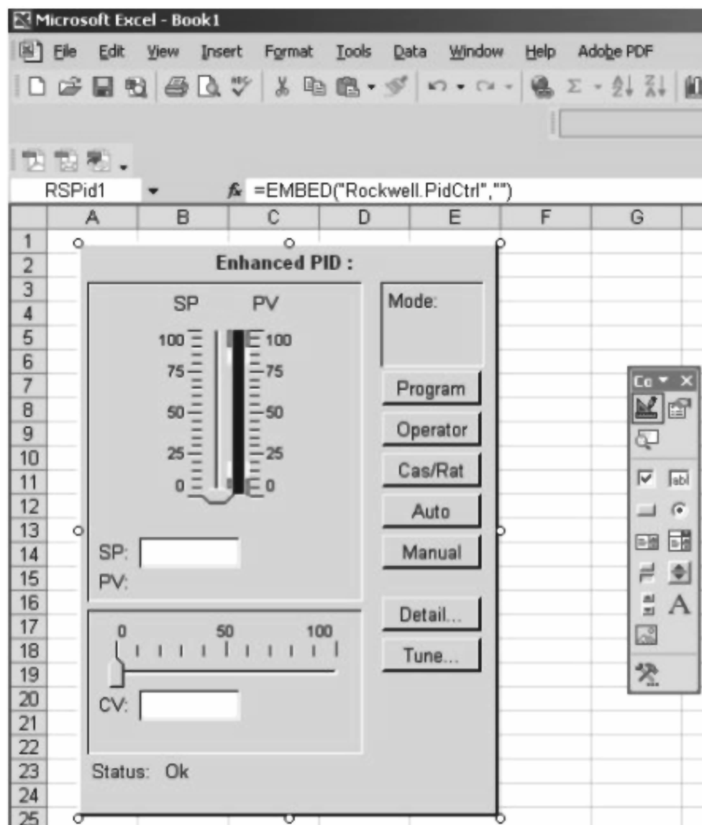


图 17-4 添加“Logix 5000 PIDE 面板控件”

接下来，在需要使用之前定义的 RSLinx 主题，将此面板链接到控制器中的 PIDE 指令。

四、在 PIDE 面板中配置通信

1. 在 Excel 中，右键单击 PIDE 面板，然后选择“Logix 5000 PIDE 面板控件对象 > 属性” （Logix 5000 PIDE Faceplate Control Object > Properties）。

2. 出现图 17-5 后，单击标签条目旁边的省略号。这将启动标签浏览器。

3. 展开主题（Language _ Example）旁边的 + 号，浏览在线标签。

4. 在在线标签下，展开程序：FB_Prog 并选择 PIDE_01 标签，然后单击“确定”（OK）。

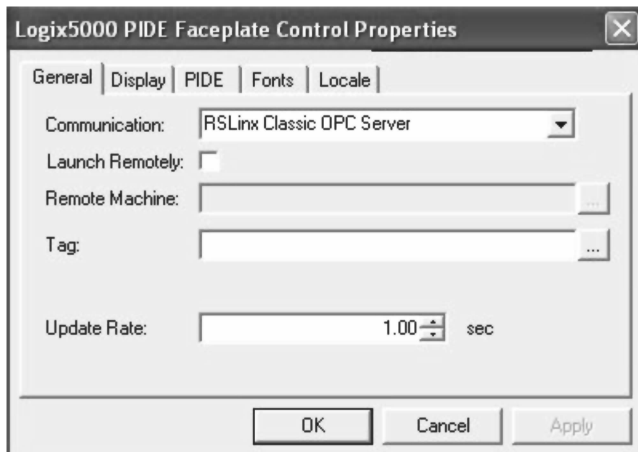


图 17-5 Logix 5000 PIDE 面板控件对象

5. 确认的配置如图 17-6 所示之后，单击“应用”（Apply）接受所做更改。单击“确定”（OK）。

6. 在控件工具箱中选择三角板图标，退出设计模式。

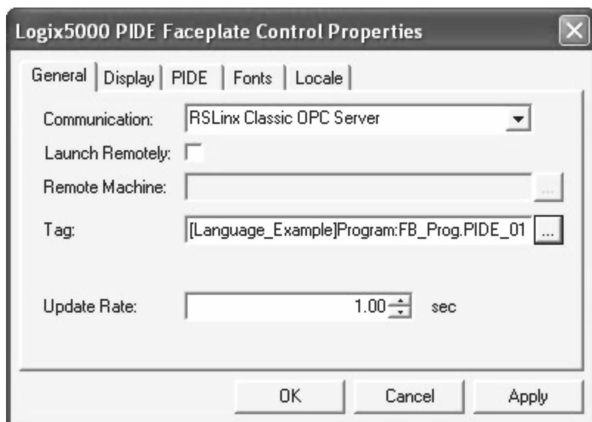


图 17-6　确认配置

五、练习使用 PIDE 面板

1. 当 PIDE 处于手动模式时，将 SP 更改为 30，可以在 SP 编辑字段中手动输入该值，或者使用垂直滑块。

2. 单击“自动”（Auto）按钮转到自动模式。

3. 观察面板中和 RSLogix 5000 中过程变量（PV）的变化情况。

4. 回到 RSLogix 5000 项目中，在趋势中单击“运行”（Run）按钮，然后在练习使用 PIDE 的同时观察过程变化情况。

5. 将 SP 更改为 0 并观察趋势。PV 应下降为 0。

6. 在 RSLogix 5000 和 Microsoft Excel 的工具栏菜单中分别选择“文件 > 退出”（File > Exit），退出项目并且不保存任何更改。

六、实验小结

在本实验中，已完成以下任务：

1）运行 RSLogix 5000 并创建在本实验练习期间将用到的项目；

2）查看控制器项目管理器，为进行实验练习的下一环节作准备；

3）创建一个名为 FB_Task 的周期性任务，将其配置为每 100ms 执行一次；

4）创建一个名为 FB_Prog 的程序，安排该程序作为 FB_Task 的一部分来执行；

5）创建一个名为 FB_Routine 的功能块例程；

6）安排 FB_Routine 作为程序 FB_Prog 的主例程来执行；

7）创建功能块回路模拟；

8）PIDE 自整定；

9）创建趋势来观察过程变量变化情况；

10）通过 Active X 面板练习使用 PIDE。

第四部分　用户自定义指令 AOI

实验 18：向 PowerFlex 设备中写入参数

一、实验目的

创建一个 AOI，用于在 PowerFlex 变频器中写入参数。将逐步完成 AOI 的配置，了解 AOI 的定义。

二、打开文件并根据工作站进行调整

1. 双击桌面上的 RSLogix 5000 图标打开 RSLogix5000。

2. 单击按钮，打开名为“mill_start. ACD”的文件夹，位置在 C：\ HOTT \ AOI & Tools。

3. 在 I/O 配置树中，右键单击 I/O 树中名为“mill_drive”的 PowerFlex 70-EC 变频器，然后选择“属性”（Properties）如图 18-1 所示。

4. 根据工作站（172. 16. XX. 7）中变频器的情况更改 IP 地址，其中，XX 是的工作站编号。

5. 单击“确定”（OK）。

6. 现在已拥有需要的所有 I/O，该对 AOI 进行参数化了。

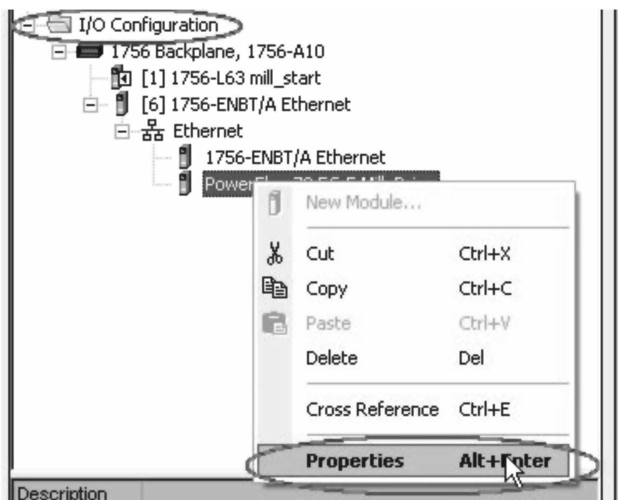


图 18-1　PowerFlex 属性菜单

7. 选择“文件→另存为”（File→Save as）”，在 C：\ HOTT \ AOI & Tools 文件夹下将该文件另存为“mill_code. ACD”。

三、导入用户自定义指令

1. 在这部分实验中，将展示一种快速导入 AOI 的简便方法。可以通过从已有项目中执行“拖放”操作来导入 AOI。在此例中，将通过导入 . L5X 文件来导入 AOI。

在项目树中，右键单击“用户自定义指令”（Add-On Instructions）文件夹并选择“导入用户自定义指令”（Import Add-On Instruction），如图 18-2 所示。

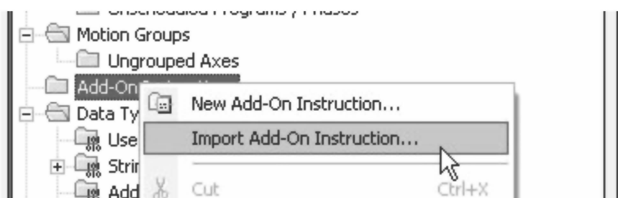


图 18-2　用户自定义菜单

2. 如图 18-3 所示，从 C：\ HOTT \ AOI & Tools 文件夹中选择“Drive_Milling_Position. L5X”文件，然后单击“导入”（Import）。

3. 将在“用户自定义指令”

（Add-On Instruction）文件夹中看到新指令，如图 18-4 所示。

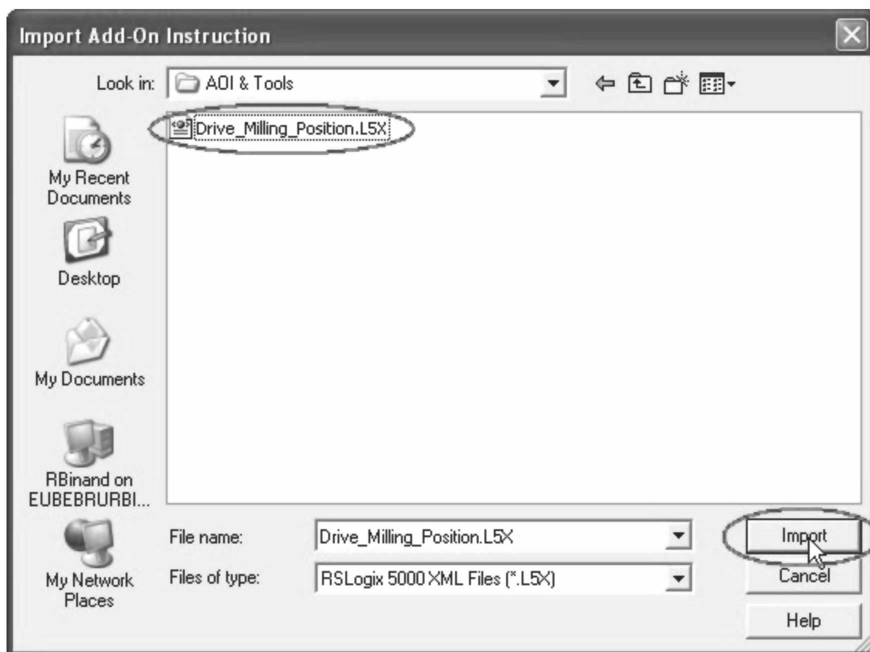


图 18-3　导入窗口

四、用户自定义指令定义

1. 现在来看一下，刚才导入了什么。右键单击 Drive_Milling_Position AOI 并选择“打开定义”（Open Definition），如图 18-5 所示。

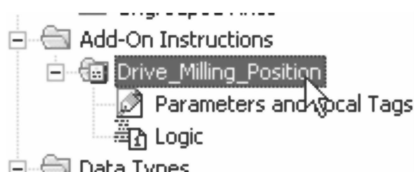


图 18-4　项目列表



图 18-5　自定义菜单

2. 将看到以下窗口。现在查看的内容被称作：“AOI 定义”，如图 18-6 所示。

参考信息

“常规”（General）选项卡包含最初创建指令之后的信息。可以使用这个选项卡来更新信息。说明、版本、版本注释和厂家信息会被复制到指令的客户帮助文件中。版本无法由软件自动管理。由负责定义如何使用软件以及何时更新。

梯形图、功能块图和结构化文本均可用于用户自定义指令逻辑。如上面所示，此 AOI 是使用结构化文本（ST）编程的。

RSLogix 5000 软件支持的每一种编程语言都针对不同类型的应用和编程方式。一般来说，梯形图执行简单布尔逻辑、定时器和计数器的速度最快。如果利用了功能块图和结构化文本中较高级的过程及变频器指令，则这两种语言可能会更加高效。

无法比较用不同编程语言编写的相同用户自定义指令的执行时间。不同语言的执行和编译存在本质差别。

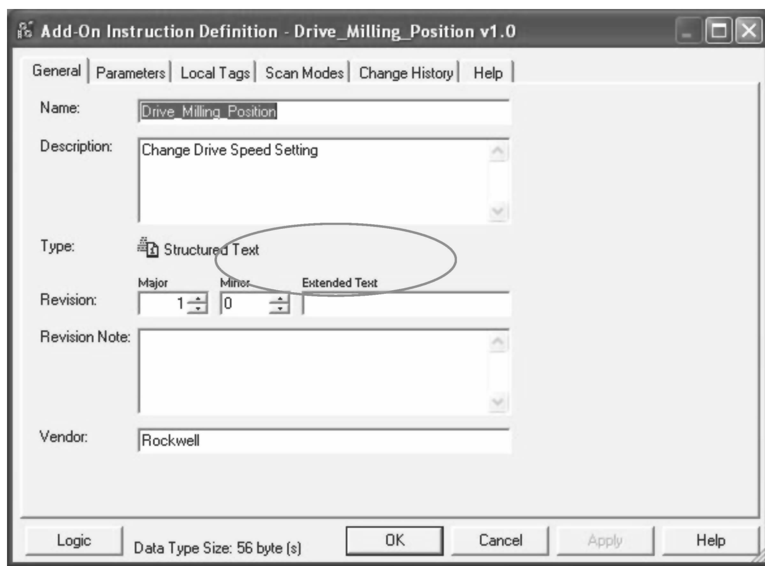
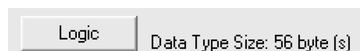


图 18-6　AOI 定义窗口

3. 请看窗口的左下角。会看到“逻辑”（Logic）按钮以及旁边的“数据类型长度”（Data Type Size）。



用户自定义指令有一个基本逻辑例程，定义了指令执行时的行为。此逻辑例程与项目中的其他例程一样，对长度没有额外限制。支持的最大数据实例（包括输入、输出和局部标签）为2MB。

4. 让了解一下 AOI 在做什么。单击“逻辑”（Logic）按钮。将看到 ST 编辑器，其中包含有 AOI 中封装的所有代码，如图 18-7 所示。

为了向 PowerFlex 变频器写入参数，封装代码使用了消息指令（MSG_Configuration）。当机器位于正确位置时，消息指令会向变频器中写入一个数据数组（MSG_Data，数据类型为 INT[36]），数组中包含需要设置的参数编号和值。发送给变频器的参数为最大频率、最大速度、超速限制值、加减速时间以及方向模式。

消息是“Set_Attributes_Scattered”类型的，可在一条消息中写入多个参数。

阅读与图 18-8 类似的部分代码：

在 STX 中，将 MSG 指令触发仅一次扫描周期是很重要的，通常情况下，指令始终都会执行。接下来的两行使用了 STX 中的单触发技术来检测发送到变频器中的消息是何时执行的。最后一行设置指令的“ER”位，并让指令在第一行后停止执行（“transition”）。

5. 关闭结构化文本编辑器并回到“AOI 定义”（AOI Definition）窗口。现在选择“参数”（Parameters）选项卡，如图 18-9 所示。



图 18-7　ST 编辑器窗口

这些参数定义了指令接口；使用时，指令的外观显示。参数顺序定义了指令调用时所显示参数的顺序。用户自定义指令参数用法和数据类型选项如下：

```
if trigger then MSG(cfgMSG); trigger:=0; end_if;
if cfgMSG.DN and NOT DN_copy then step:=step+10; end_if;
DN_copy [:=] cfgMSG.DN;
if cfgMSG.ER then step := 90;ER:=1;transition:=0;end_if;
```

图 18-8　消息代码

1）输入参数：向用户自定义指令传送值。必须是 SINT、INT、DINT、REAL 或

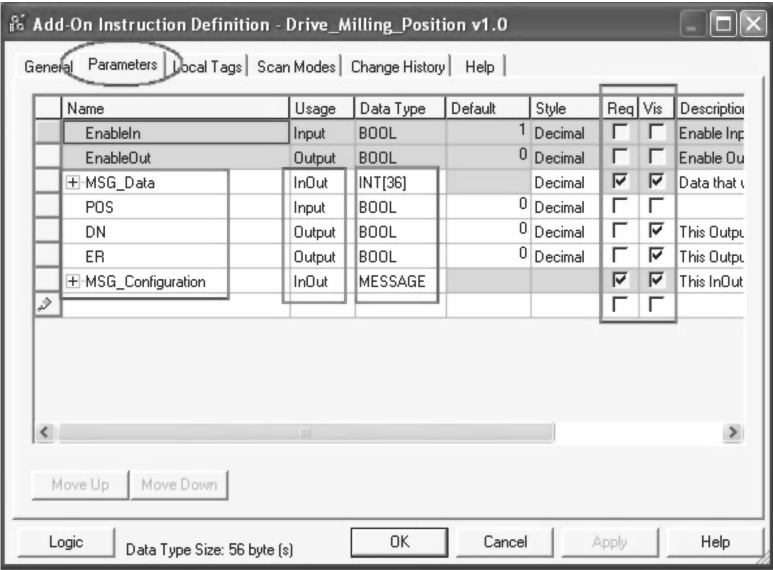


图 18-9　参数选项卡

BOOL 数据类型。在该示例中，POS 被定义为一个输入参数。POS 可以从 AOI 外部访问。在用户的应用中，POS 将会在下一个 AOI 中用到，即根据机器的位置改变变频器速度的时候。

2）输出参数：向用户自定义指令外传送值。必须是 SINT、INT、DINT、REAL 或 BOOL 数据类型。在该示例中，DN 和 ER 被用作输出参数。它们将显示在 AOI 的右侧，为用户提供指令执行情况的反馈信息。

3）读写参数：向用户自定义指令内传送外部引用。可以是任意数据类型，包括结构和数组。在该示例中，Message 标签需要通过指令中的引用来传送，在程序中插入 AOI 后将完成 Message 配置。MSG_Data 是一个数组，因此该标签的用法只能是“读写”。

一个只含有输入参数（EnableOut 除外）的指令在梯形图中会被视为输入指令。EnableOut 参数用于梯级条件。

参考信息

可以使用必要参数来确保特定数据已被送入用户自定义指令中。必要参数必须作为自变量进行传递，以便调用指令来验证。在梯形图和结构化文本中，这可以通过为这些参数指定一个自变量标签来实现。在功能块图中，必要的输入和输出参数必须连线，而读写参数必须有自变量标签。

对输出参数来说，如果不需要经常向自变量传递参数值，但为了故障排查，又想突出显示出它的值，那么让参数可见是非常有用的。必要参数始终是可见的，而读写参数都是必要参数并且始终可见。无论被标记为必要的还是可见的，所有的输入和输出参数均可作为指令标签的成员通过程序访问。对于读写参数，可见设置始终是可见的。所有读写参数都是必要的。

如果有一个参数，用户必须给这个参数指定一个标签作为输入源或输出目标，并且不想让其成为可选的，那么请将该参数设置为“必要”的。任何必要参数都会被自动设置为可见。

用户自定义指令的参数可以通过符合正常标签范围规则的指令定义标签，在控制器编程过程中直接访问。每个用户自定义指令默认情况下都有“EnableIn”和“EnableOut”参数，这两个参数都具有符合不同语言环境（梯形图逻辑、功能块图、结构化文本）的特性。

执行基本逻辑例程时，“EnableIn”参数在任何语言环境中都为真（1）。一般来说，“EnableIn”参数不应被指令定义中的基本逻辑例程引用。

默认情况下，“EnableOut”会遵循“EnableIn”的状态，但是用户逻辑可以强制该参数的状态。

在梯形图逻辑环境下，“EnableIn”参数反映了输入指令的梯级状态。例如，如果指令前的梯级状态为真（1），则“EnableIn”为真，并且将执行指令的基本逻辑例程。

在功能块环境下，用户可以通过引脚连接来操作“EnableIn”参数。如果没有连接，则当指令开始执行时“EnableIn”参数会被设置为“真”（1），并会执行指令的基本逻辑例程。如果参数上的连接为“假”（0），则不会执行指令的基本逻辑例程。写入 EnableIn 参数的其他引用不会影响该参数的状态。只有此参数输入引脚上的连接才能将其强制为“假”（0）。

在结构化文本环境中，默认情况下“EnableIn”参数始终设置为“真”（1）。用户无法对结构化文本中“EnableIn”参数的状态造成影响。

6. 现在选择“帮助”（Help）选项卡。RSLogix 5000 软件会使用指令说明、版本注释和参数说明，为用户自定义指令自动建立帮助。可以通过创建有价值的说明来帮助使用该指令的用户。此外，还可以通过使用扩展说明字段将自己的自定义文本添加进帮助。

创建用户自定义指令并将信息添加进说明字段时，会自动生成用户自定义指令帮助。如要提供更多帮助文档，还可以将其输入用户自定义指令定义编辑器的帮助选项卡中。可在指令浏览器中找到指令帮助，也可在语言编辑器中存在该指令的调用时按 F1 键来打开。

用户自定义指令帮助让用户在使用指令的时候更方便地获得所需的帮助。

编写说明时，请记住以下准则。

1）使用简短、易懂的语言；

2）所写内容应简要直接；

3）列举简单的示例；

4）确认字段输入内容，如图 18-10 所示。

Available Languages

Relay Ladder

Function Block

Structured Text

Drive_Milling_Position(MSG_Data, MSG_Configuration);

Parameters

Required	Name	Data Type	Usage	Description
X	Drive_Milling_Position	Drive_Milling_Position	InOut	Change Drive Speed Setting
	EnableIn	BOOL	Input	
	EnableOut	BOOL	Output	
X	MSG_Data	INT	InOut	Data that will be used in the Message to change the drive settings
	POS	BOOL	Input	This Input parameter is used in the other AOI inorder to change the Drive speed regarding the position of the machine
	DN	BOOL	Output	This Output Parameter is visible in order to show the right execution of the AOI
	ER	BOOL	Output	This Output Parameter is visible in order to show if an error occured in the execution of the AOI
X	MSG_Configuration	MESSAGE	InOut	This InOut parameter will allow you to configure the message instruction (like the comm path to the drive,...)

Extended Description

This instruction is using a message instruction to set the parameters in the drives. The controller will execute the message and writes an array of integer that contains the parameter number and its value. The code encapsulated in the AOI will also make sure that the data transfer is done in a limited amount of time.

Execution

Condition	Description
EnableIn is false	
EnableIn is true	

Revision v1.0 Notes

Original AOI version

图 18-10　编写说明

7. 选择“局部标签”（Local Tags）选项卡，如图 18-11 所示。

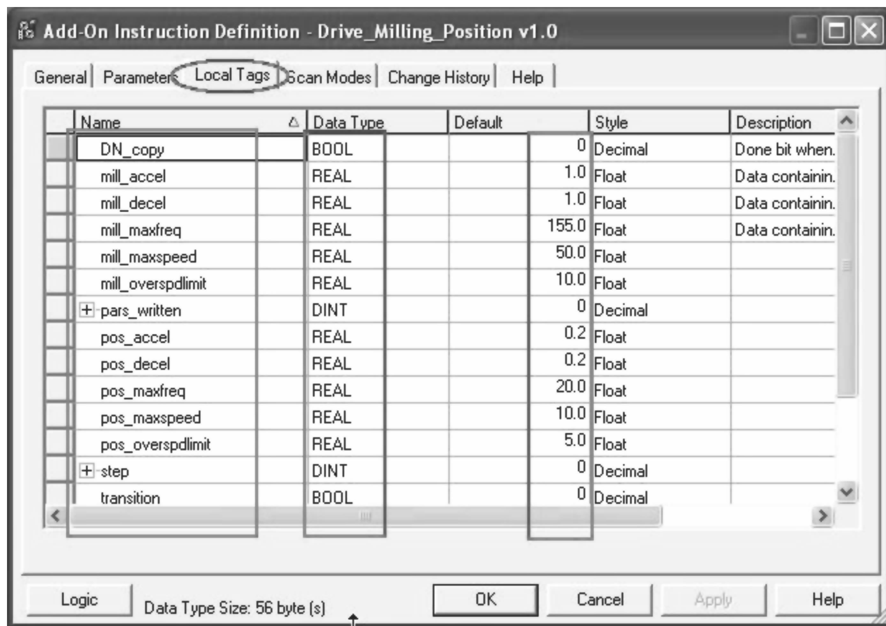


图 18-11　局部标签选项卡

参考信息

局部标签包含了用户自定义指令中使用的，但是又不想暴露给客户的数据。局部标签不会显示在用户自定义指令的数据结构中，因为它们是隐藏成员。通过指定局部标签的名称作为用户自定义指令型标签中的成员，即可从 HMI 访问局部标签值。

以下数据类型不可用于局部标签-ALARM_ANALOG、ALARM_DIGITAL、MESSAGE 或者其他运动控制数据类型，例如 Axis（轴）或 MOTION_GROUP。要在用户自定义指令中使用这几类标签，请定义一个读写参数。与用户数据类型一样，局部标签同样仅限一维数组。

当创建用户自定义指令数据类型的标签，或在指令定义编辑器中向用户自定义指令定义添加新标签时，会将默认值从用户自定义指令定义加载到标签中。

用户自定义指令定义中的局部标签值是默认值。更改一个现有的局部标签不会影响已经存在的用户自定义指令的实例。要确定某个值是否已应用到指令中的所有实例，可立即将数据分配给用户自定义指令的代码中的标签。

8. 选择“扫描模式”（Scan Modes）选项卡，如图 18-12 所示。

参考信息

指令定义编辑器中的“扫描模式”（Scan Modes）选项卡可用于创建或启动例程的执行，包括三种扫描模式：Prescan、Postscan 和 EnableInFalse。

为使用户自定义指令拥有与内置指令一样的灵活性，可以对可选的扫描模式例程进行

配置，从而能够全面地定义指令的行为。用户自定义指令中开始并不存在扫描模式例程，因为它们是可选的，必须由客户根据指令的需要来创建。

对于指令的预期操作而言，在未创建可选扫描模式的情况下执行用户自定义指令的这种默认行为可能足以满足要求了。如果未定义可选的操作模式，则用户自定义指令的默认行为如下：

真：执行用户自定义指令的主逻辑例程。

假：不执行用户自定义指令的逻辑，并且不会写入任何输出。输入参数会传送来自相应自变量的值。

预扫描：在预扫描（Prescan）模式下执行用户自定义指令的主逻辑例程。会传送所有必要的输入、输出参数的值。

后扫描：在后扫描（Postscan）模式下执行用户自定义指令的主逻辑例程。

对于每一种扫描模式，都可以定义一个专门为该扫描模式编写的例程，并可进行配置以便在该模式下执行。

真：执行用户自定义指令的主逻辑例程（非可选）。

假：当发生指令扫描错误时，通常由 EnableIn False 例程代替主逻辑执行。会传送所有必要的（或 FBD 中连接的）输入、输出参数的值。

预扫描：通常在主逻辑例程的预扫描执行结束后执行本 AOI 的预扫描例程。会传送所有必要的输入、输出参数的值。

后扫描：通常在主逻辑例程的后扫描执行结束后执行本 AOI 的后扫描例程。

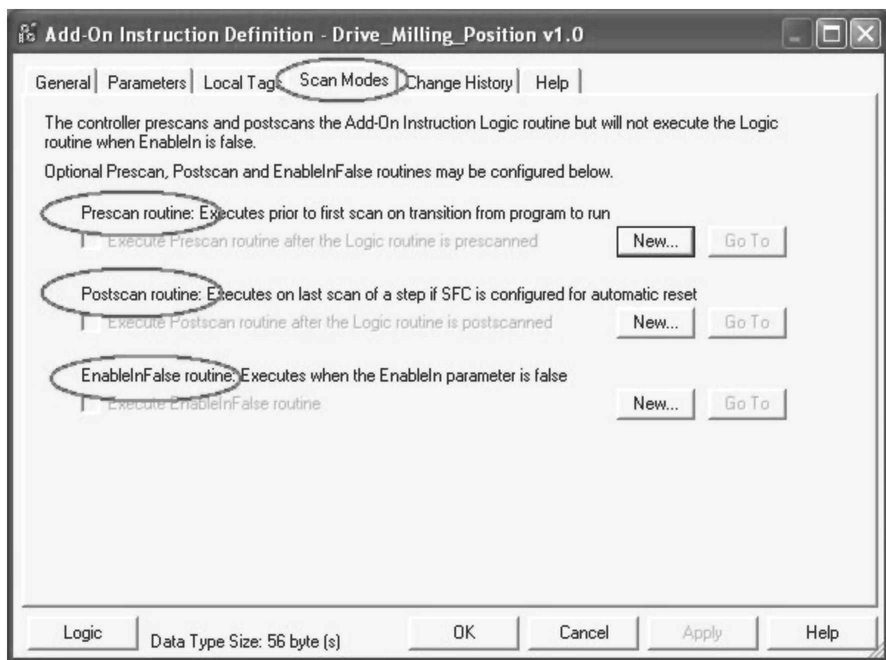


图 18-12　扫描模式选项卡

五、创建 EnableInFalse 例程

1. 向变频器写入参数的操作不一定是连续的，而是根据需要。因此，需要确保 EnableIn 参

数为真时指令只执行一次，如前文所述。

为了再次执行逻辑，要在指令未启用时使用 EnableInFalse 例程来“复位”指令，从而为下一次触发做好准备。

2. 选择“扫描模式”（Scan modes）选项卡并单击“新建”（New）按钮来创建一个 EnableInFalse 例程，如图 18-13 所示。

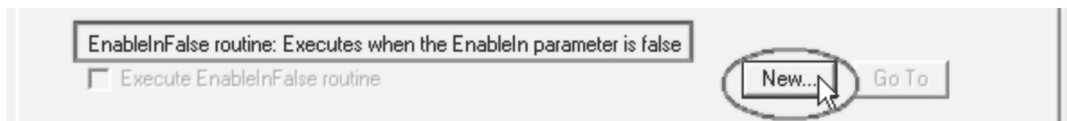


图 18-13 新建按钮

3. 确保类型已设置为结构化文本，并且“打开例程”（Open Routine）复选框处于未选中状态，然后单击“确定”（OK），如图 18-14 所示。

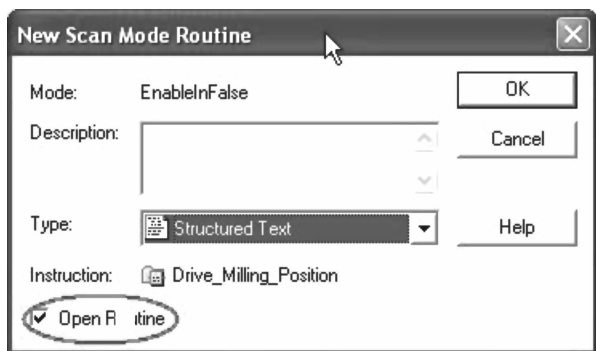


图 18-14 新建扫描模式例程

4. 现在，在 AOI 定义扫描模式选项卡中，将看到“执行 EnableinFalse 例程”（Execute EnableinFalse routine）复选框已被选中，而“转到”（Go To）按钮不再灰显，如图 18-15 所示。

5. 现在单击“应用”（Apply），会看到一个 EnableInFalse 例程，如图 18-16 所示。

6. 打开“逻辑”（Logic）例程 Logic 并查看以下代码（程序中的第一行代码），如图 18-17 所示。

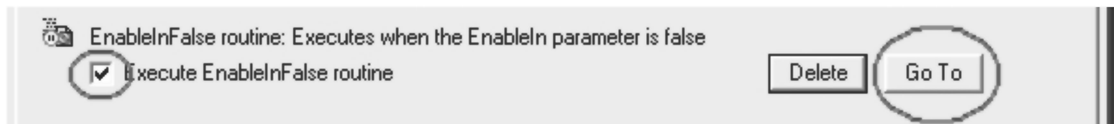


图 18-15 Go To 按钮

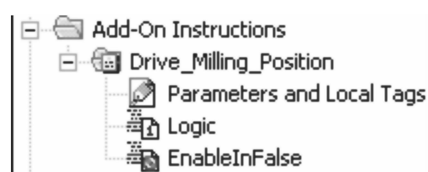


图 18-16 项目列表

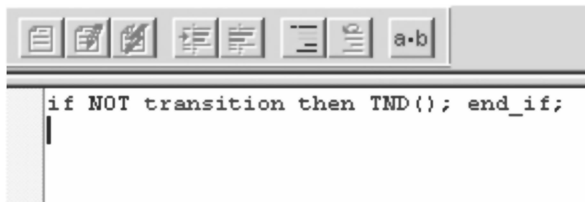


图 18-17 例程代码

这段代码的意思是，当 Transition 等于 0 时，处理器不会执行这行后面的代码。现在，需要确保 EnableInFalse 例程会在下次执行中将 Transition 设置为 1。

7. 关闭 ST 编辑器并双击 EnableInFalse EnableInFalse 逻辑图标，输入以下代码，如图 18-18 所示。

在主指令逻辑中“step”标签将用于为 Action 排序。

```
transition :=1;step:=10;          //arm instruction for next false-to-true transition
```

图 18-18　EnableInFalse 代码

8. 保存项目并关闭语言编辑器，关闭 AOI 定义窗口。

9. 第一个用户自定义指令已完成了。但请记住，如果要在程序中使用，则必须配置 Message 指令。

实验 19：AOI 用来控制 PowerFlex 设备

一、实验目的

在这部分实验中，将创建一个 AOI，用于控制一台 PowerFlex 变频器。这个 AOI 还会触发之前创建的 AOI。机器的位置设置好后，立刻改变变频器的速度。

二、创建用户自定义指令

1. 在项目树中，右键单击“用户自定义指令”（Add-On Instructions）文件夹并选择“新建用户自定义指令”（New Add-On Instruction），如图 19-1 所示。

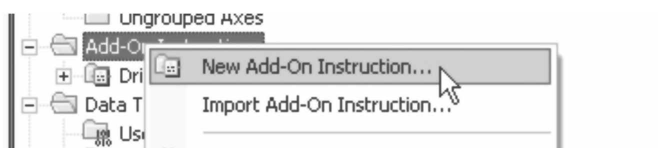


图 19-1　新建用户自定义指令菜单

2. 在新窗口中输入以下信息（名称：Drive_Milling_Control，版本注释：原始 AOI 版本），并确保 AOI 的类型为“梯形图”（Ladder Diagram），如图 19-2 所示。

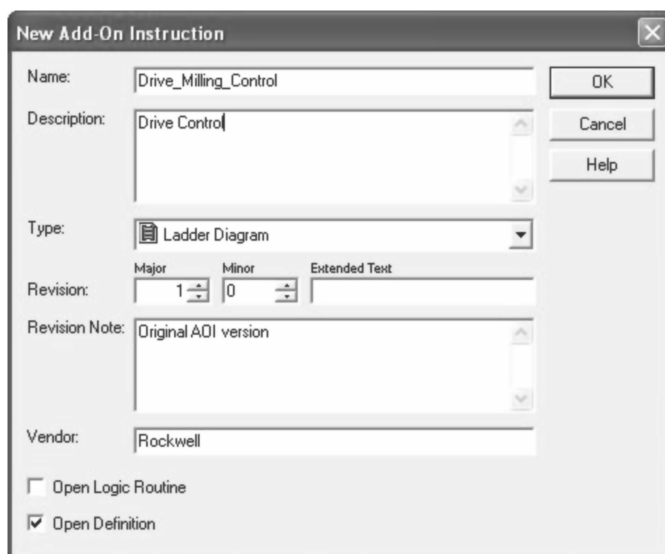


图 19-2　新建用户自定义指令窗口

3. 单击“确定”（OK）。打开“参数”（Parameters）选项卡。

4. 创建以下参数。如图 19-3 所见，作为读写参数也可以使用模块定义类型。

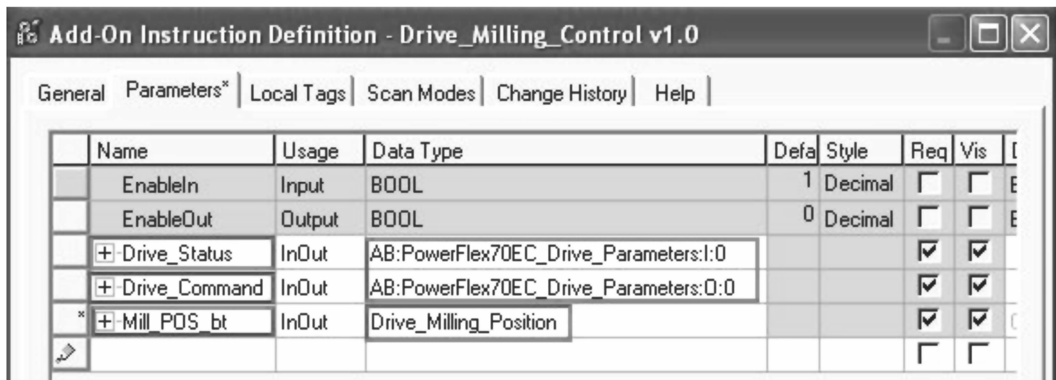


图 19-3　参数选项卡

当 PowerFlex 70EC 已经在 I/O 配置中配置完毕后，通过“Drive_Status”和“Drive_Command”参数，指令可以直接向 RSLogix5000 自动创建的模块定义标签读取或写入数据。

“Mill_POS_bt”读写参数属于之前创建的 AOI 的类型。这样此 AOI 可以监控其他 AOI。

5. 单击“应用”（Apply）。

6. 对于其他参数和标签，请选择“工具→导入”（Tools→Import），然后从“Mill_Start-PF_Mill_Ctrl－Tags. CSV”导入标签，如图 19-4 所示。

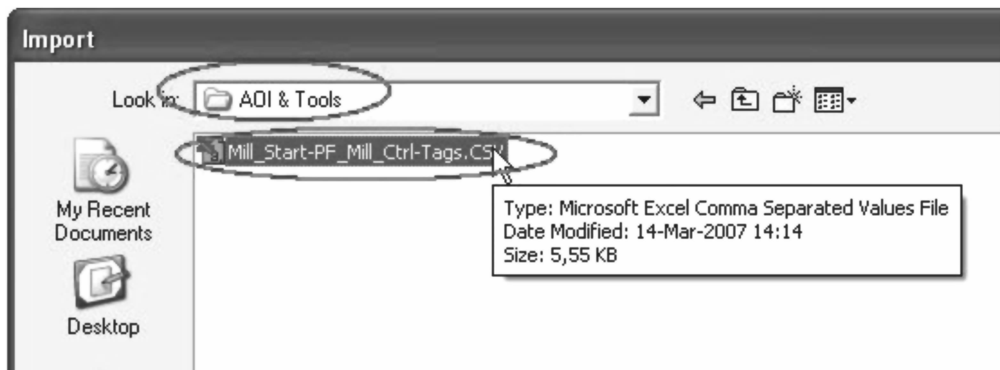


图 19-4　导入对话框

确保导入操作顺利完成并且在错误窗口中看到以下消息：

```
Complete – 0 error(s), 0 warning(s)
```

查看输入和输出参数。没有必要参数，但有一些是可见的参数。这样无需针对 HMI 的额外代码，即可使用 HMI 对数据块进行监控。

7. 选择“帮助”（Help）选项卡并检查指令将如何显示，如图 19-5 所示。

“entrystatus”参数用于防止指令试图对不在网络中的变频器进行控制或配置。

需要使用 GSV 指令来获取变频器的 Entrystatus（I/O 状态）。由于 GSV 会引用一些外部信息，因此它不能在 AOI 内。

8. 保存项目。

9. 双击 Logic 以打开指令逻辑并输入以下逻辑，如图 19-6 所示。

通过操纵系统定义的“EnableOut”参数，可以控制同一梯级中某指令右侧的指令是否执行。

10. 同以前一样，无需花费大量时间输入代码。从“C：\ HOTT \ AOI & Tools”文件夹打开名为“PF_Mill_Control_Logic.txt”的文件，按 CTRL-A 和 CTRL-C。

11. 在项目中，双击“结束”(End) (End) 梯级。

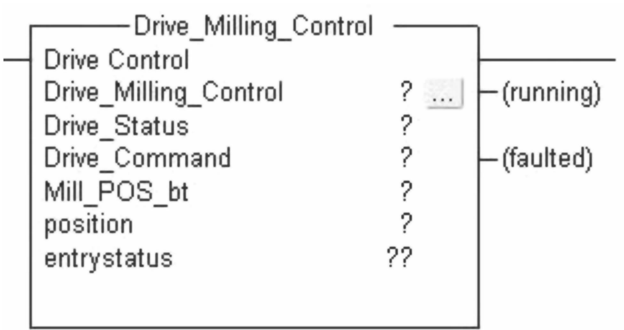


图 19-5　显示界面

12. 在梯形图编辑器窗口上选择“中性文本”(Neutral text)，如图 19-7 所示。



图 19-6　逻辑指令

13. 右键单击旁边的字段并选择“粘贴”(Paste)，如图 19-8 所示。



图 19-7　文本选项

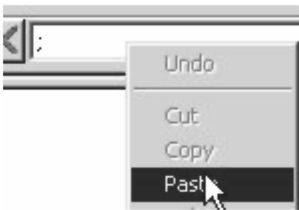


图 19-8　粘贴选项

14. 按“回车”(Enter) 后会出现代码。

15. 选择“中性文本”(Neutral text) 可以将许多梯级粘贴到 RSLogix5000 中。对于通过记事本操作或用 excel 生成的代码，这种方法可以高效地进行粘贴。

16. 浏览刚才创建的代码。大部分是控制变频器的简单代码，梯级 5-9。最后几个梯级中的“更换”(change) 位会触发 AOI，使其向变频器写入参数。

17. 如果有的话，请删除末尾的空梯级（梯级 18）。

18. 右键单击指令并选择“验证”(Verify)，如图 19-9 所示。

同样，如果出现任何错误，请向讲师求助。

19. 保存 项目。

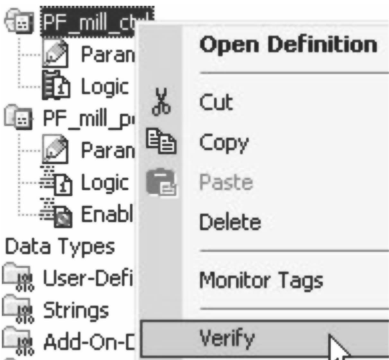


图 19-9　右键菜单

实验 20：在代码中使用用户自定义指令

一、实验目的

使用不同的技术将 AOI 添加到程序中，同时还会了解到如何管理 AOI 的不同版本。

二、在添加 AOI 之前

1. 在使用 AOI 之前，需要添加一些梯级代码来获取有关控制器与变频器连接状态的信息。

展开“主程序”（MainProgram）MainProgram 并双击“主例程”（Mainroutine）MainRoutine 输入以下逻辑，如图 20-1 所示。

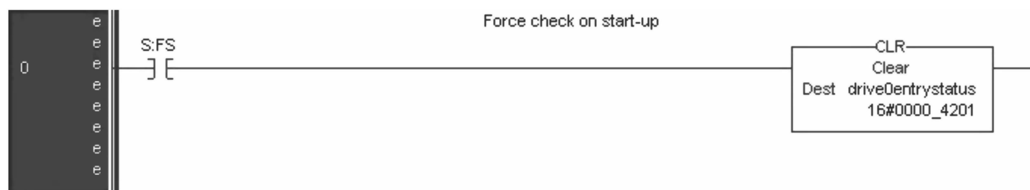


图 20-1 逻辑指令

2. 右键单击“drive0entrystatus”并选择“New“drive0entrystatus””，如图 20-2 所示。

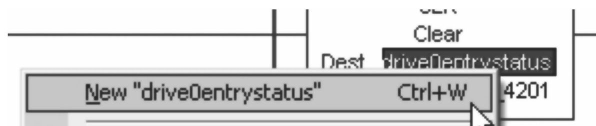


图 20-2 新建“drive0entrystatus”命令

根据图 20-3 所示截图设置类型、数据类型、范围和格式。

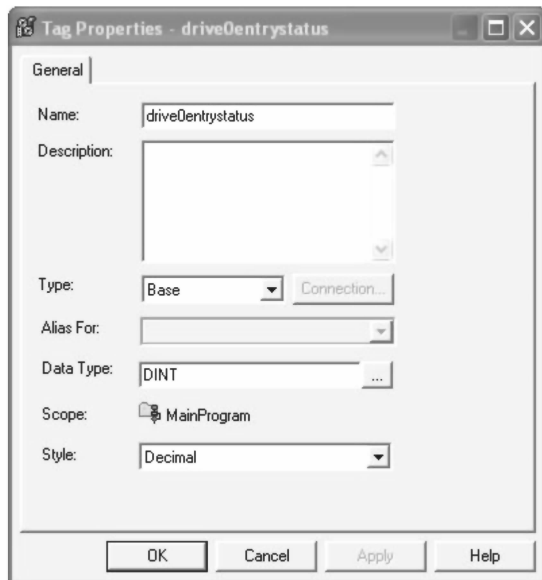
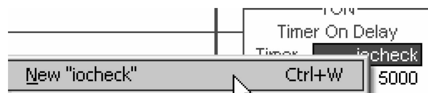


图 20-3 属性对话框

3. 刚才创建的梯级可确保控制器进入运行模式时，会复位变频器的连接状态，使得必须先连接才能与 Powerflex 设备进行通信。

4. 现在，输入以下逻辑，如图 20-4 所示。

5. 右键单击“iocheck”并选择



“New“iocheck””。请依照图 20-5 所示的屏幕截图设置并单击“确定”（OK）。

6. 现在输入以下梯级，如图 20-6 所示。

由于此梯级中需要的所有标签都已存在，因此不需要创建任何标签。GSV 在 I/O 树中引用了 Powerflex 变频器。由于这是指令外部的对象，因此这段代码不能在 AOI 内部。

创建的这三个梯级每 5s 监视一次 Powerflex 的连接状态。

Drive_Milling_Control AOI 会利用连接状态信息，以便仅在 Powerflex 连接的时候尝试操作。

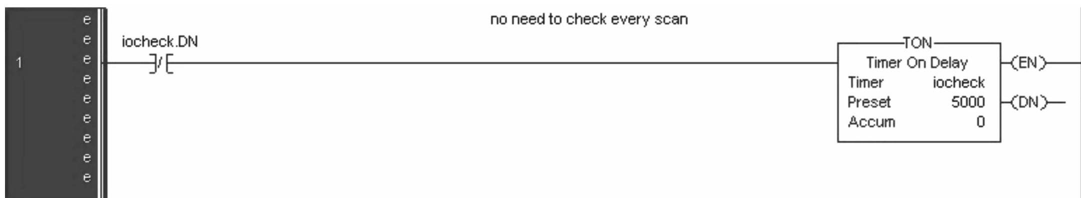


图 20-4 逻辑指令

三、向程序添加 AOI

1. 现在该使用全新的指令了。在例程中使用用户自定义指令的方法与使用内置指令一样。向指令中添加调用，然后定义参数的自变量。用户自定义指令可用于任何梯形图、功能块或结构化文本语言（包括顺序功能图操作中的结构化文本）。

指令的外观符合所处的语言环境。项目中的用户自定义指令可通过任意正常指令选择工具进行访问。

指令工具栏中有一个“用户自定义指令”（Add-On）选项卡，其中列有项目中当前可用的所有用户自定义指令，如图 20-7 所示。

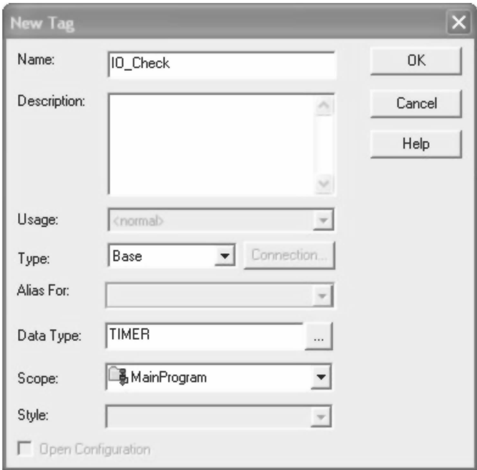


图 20-5 新建标签对话框

2. 通过点击位于指令工具栏中的 按钮，插入一个新梯级。在编辑器中的任意位置按下 Alt + Insert 键或右键单击编辑器中的逻辑。找到要添加到例程中的用户自定义指令，如图 20-8 所示。

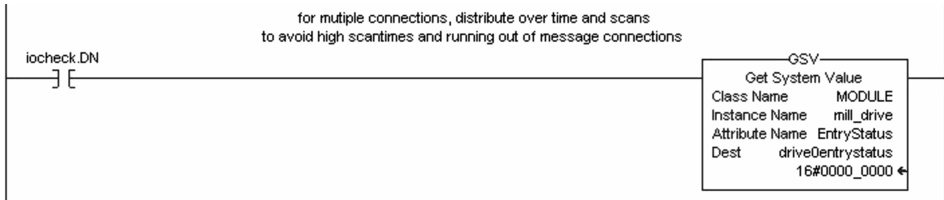


图 20-6 逻辑指令

利用“指令帮助”（Instruction Help）按钮来显示有关浏览器中任意指令的指令帮助，如图 20-9 所示。

3. 选择“Drive_Milling_Control”用户自定义指令并单击“确定”（OK）。



图 20-7 指令工具栏

4. 用同样的方法添加第二个 AOI：“Drive_Milling_Position”。梯级应如图 20-10 所示。

5. 指令需要与标签一起使用，表 20-1 总结了一些需要了解的规范。

6. 每一个 AOI 实例都有各自的支持标签。PFMC01 是 Drive_Milling_Control AOI 的支持标签，PFMP01 是 Drive_Milling_Position AOI 的支持标签。如图 20-11 所示配置这两个 AOI，请在创建标签时注意所选择的范围。

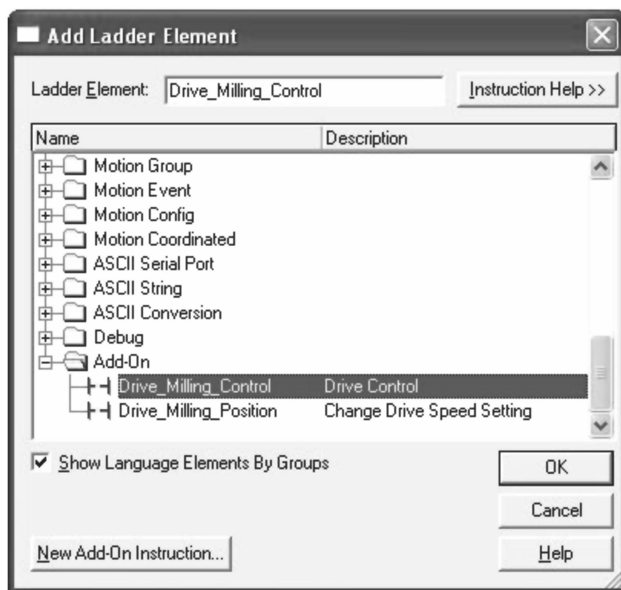


图 20-8　添加梯级对话框

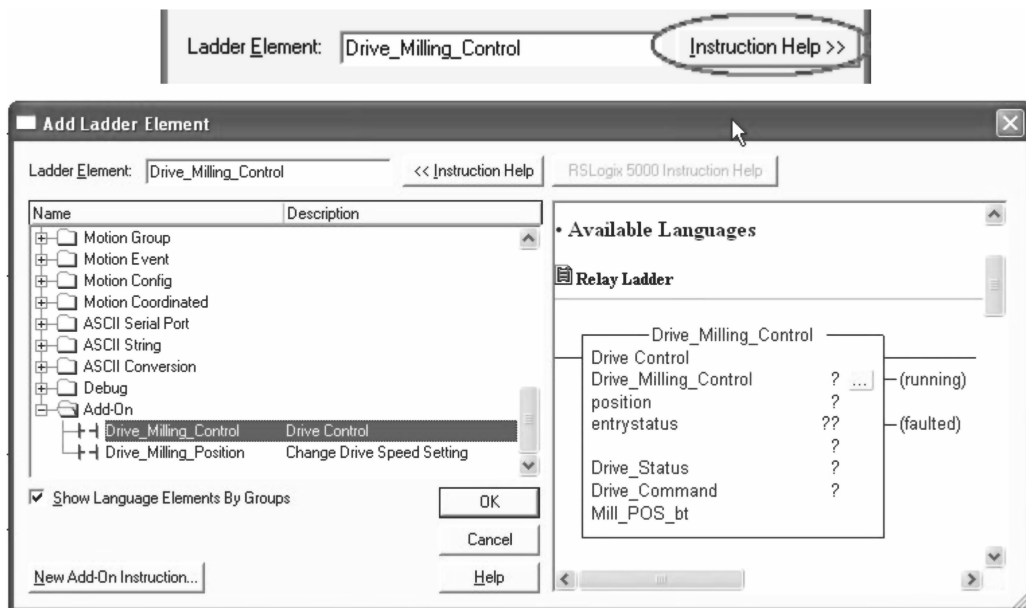


图 20-9　指令帮助对话框

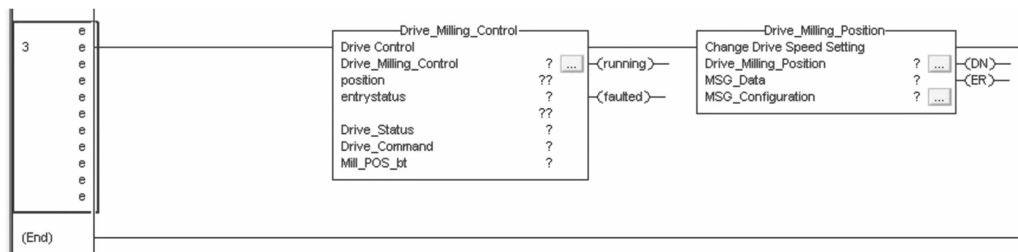


图 20-10　添加 AOI

表 20-1　指令标签使用规范

Parameter With	Description
Single question mark	This is a required Parameter. Enter a tag.
Single and double question marks	This is a required Parameter. Enter a tag.
Double question marks	This is not a required Parameter. You can either: ● leave as is and use the default value. ● if it's an input value, enter a different value.

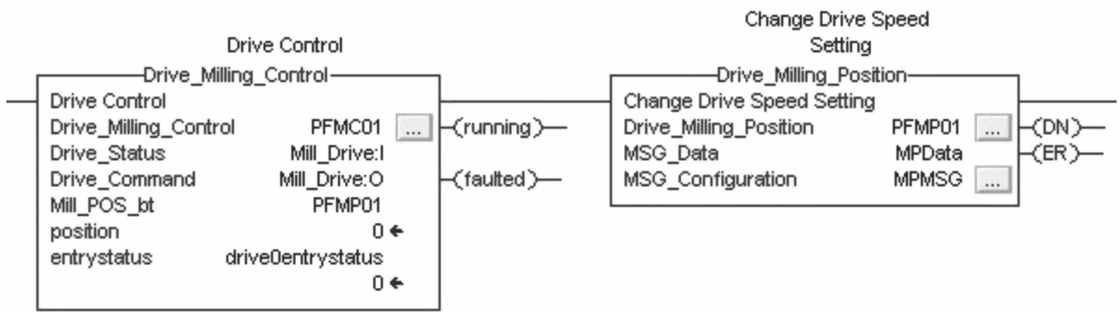


图 20-11　配置 AOI

7. 首先从 Drive_Milling_Position AOI 的配置开始。右键单击“PFMP01”并选择“New “PFMP01””，如图 20-12 所示。

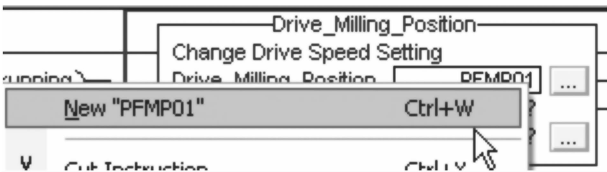


图 20-12　新建“PFMP01”命令

8. 严格依照图 20-13 所示的屏幕截图设置并单击“确定”（OK）。

由于 RSLogix5000“知道”需要正确的标签类型，因此会帮选择。

9. 因为在 AOI 中使用了消息指令，所以需要创建一个 MESSAGE 标签并配置消息指令。

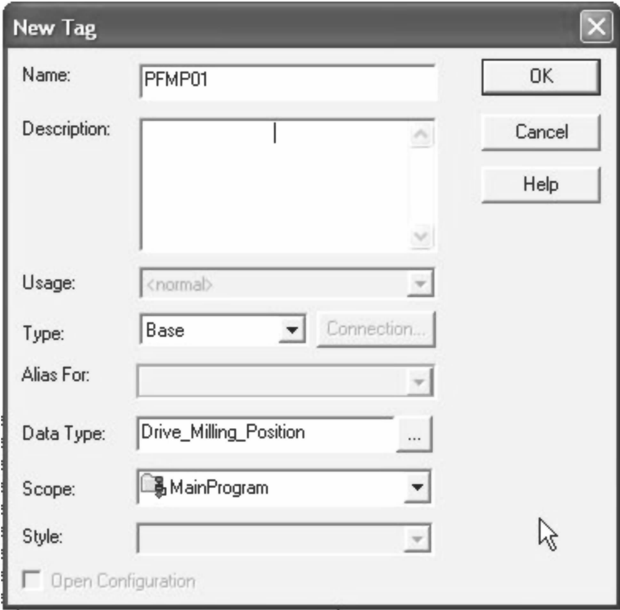


图 20-13　新建标签对话框

10. 右键单击“MPMSG”并选择“New“MPMSG”，如图 20-14 所示。



图 20-14　选择界面

11. 请依照下面的屏幕截图设置（如图 20-15 所知，消息支持标签及数据必须存储在控制器范围级别！）并单击“确定”（OK）。

12. 现在，需要对包含所有参数和参数值的标签进行配置。右键单击“MPData”并选择“New“MPData””，如图 20-16 所示。

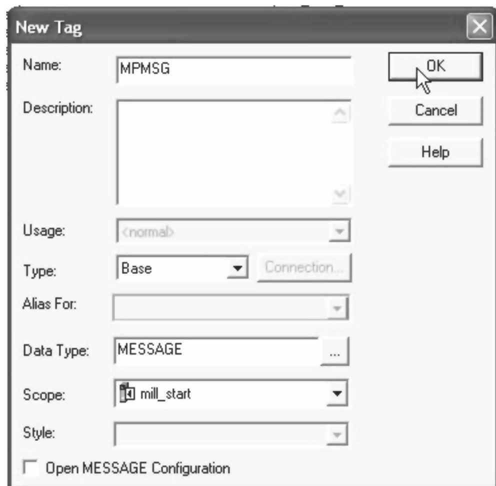


图 20-15　新建标签对话框



图 20-16　选择界面

13. 请依照图 20-17 所示的屏幕截图设置并单击“确定”（OK）。

同样，RSLogix 会自动选择正确的数据类型来匹配 AOI 的定义。

14. 对于这个块，需要做的只剩下配置消息了。单击“MPMSG”旁边的按钮，如图 20-18 所示。

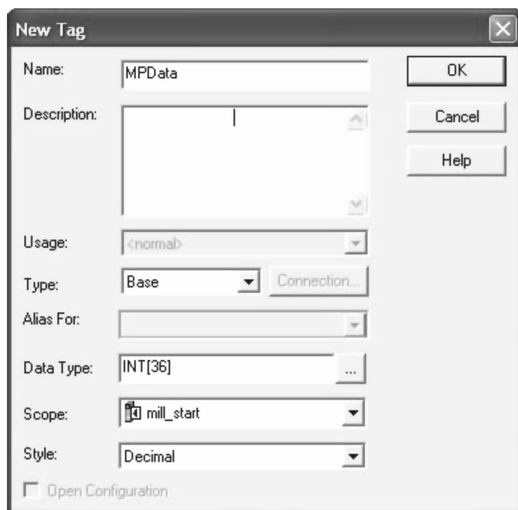


图 20-17　新建标签对话框

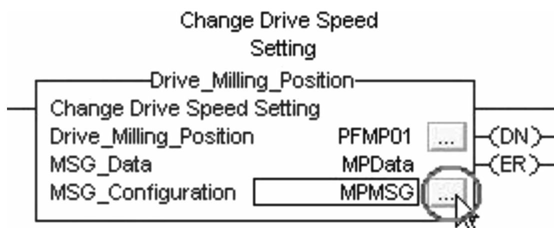


图 20-18　配置消息

可以看到消息的配置对话框，请按照图 20-19 所示屏幕截图进行配置。

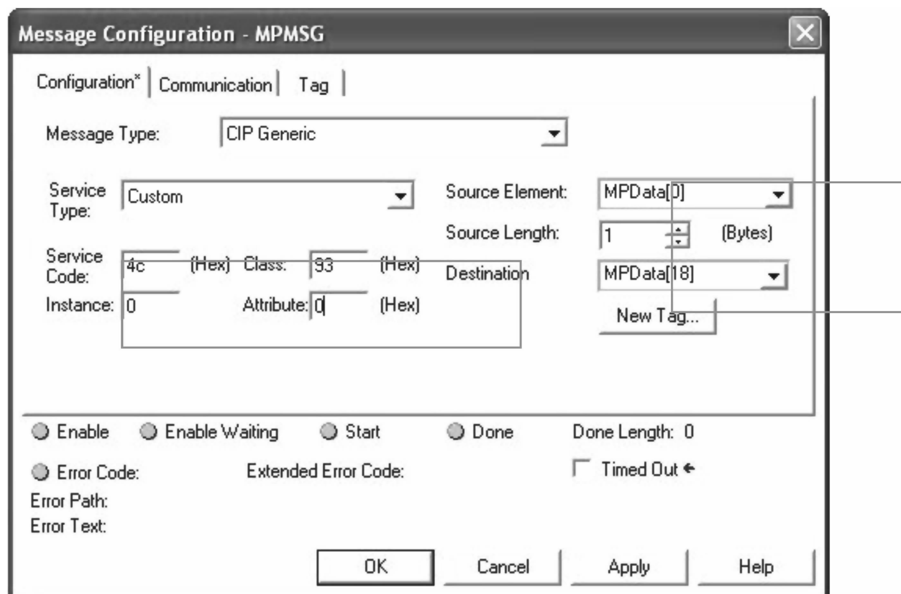


图 20-19　配置消息对话框

如前文所述，该指令需要在一条消息中写入多个参数。阅读 PowerFlex 70 的 EtherNet/IP 通信适配器（20-COMM-E）用户手册，可以轻松地找到写入多个参数所需的消息配置设置。下面是摘自该手册的概述表（见表 20-2）。

表 20-2　消息配置设置列表

Configuration	Value	Description
Service Type	Custom	Required for scattered messages
Service Code	4C	Code for Get Attributes Scattered
Class	93(Hex.)	DPI Parameter Object
Instance	0(Dec.)	Required for scattered messages
Attribute	0(Hex.)	Required for scattered messages
Source Element	SetScatteredRequestData	Controller tag for request data
Source Length	30Bytes	Fifteen 16-bit words of data
Destination	SetScatteredResponseData	Controller tag for response data

15. 选择“通信”（Communication）选项卡并单击“浏览”（Browse）。在下一弹出窗口中，选择 I/O 树中的 Powerflex 变频器，并单击“确定”（OK），如图 20-20 所示。

16. 现在单击“应用”（Apply）。

17. 剩下的只有控制 AOI 的支持标签了。右键单击“PFMC01”并选择“New“PFMC01””，如图 20-21 所示。

18. 请依照图 20-22 所示的屏幕截图（注意范围!）并单击“确定”（OK）。

现在梯级左侧的“e e e”字母消失了，表明代码是正确的。

19. 来确认下项目是否已准备好进行测试。单击“校验控制器”（Verify Controller）按

钮，看到屏幕左下角显示以下文本：

Verify complete with no errors or warnings.

保存项目。

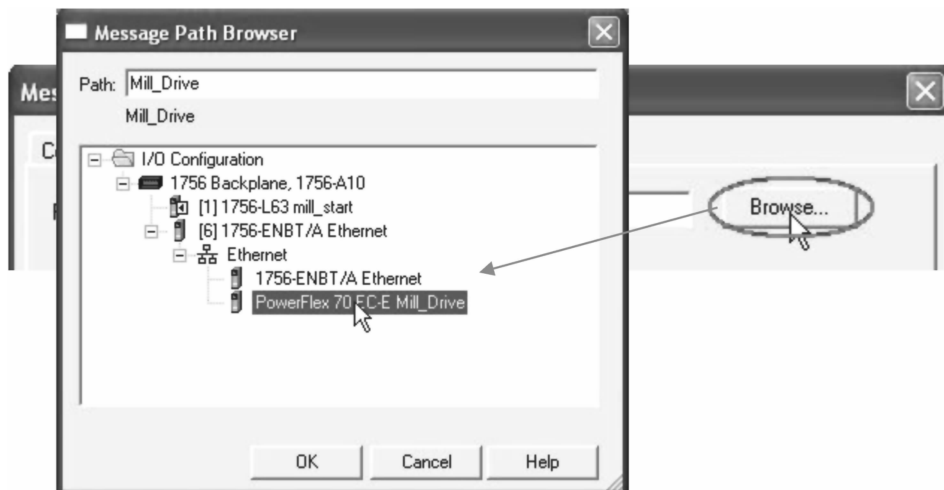


图 20-20 消息对话框



图 20-21 选择界面



图 20-22 新建标签对话框

20. 在测试应用之前，先看一下 AOI 的支持标签。在项目管理器窗口中，打开“程序标签”（Program Tags）并查看支持标签的结构（见图 20-23）。其结构都是基于配置的输入和输出参数。

MainTask
MainProgram
Program Tags
MainRoutine
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
Drive_Milling_Control
Parameters and Local Tags
Logic
Drive_Milling_Position
Data Types
User-Defined
Strings
Add-On-Defined
Predefined
Module-Defined
Trends
I/O Configuration
1756 Backplane, 1756-A10

−PFMC01	{...}	Drive_Milling_Control
PFMC01.EnableIn	1	BOOL
PFMC01.EnableOut	0	BOOL
PFMC01.position	0	BOOL
PFMC01.start	0	BOOL
PFMC01.stop	0	BOOL
PFMC01.jogfwd	0	BOOL
PFMC01.jogrev	0	BOOL
PFMC01.reset	0	BOOL
+PFMC01.entrystatus	0	DINT
PFMC01.running	0	BOOL
PFMC01.faulted	0	BOOL
−PFMP01	{...}	Drive_Milling_Position
PFMP01.EnableIn	1	BOOL
PFMP01.EnableOut	0	BOOL
PFMP01.POS	0	BOOL
PFMP01.DN	0	BOOL
PFMP01.ER	0	BOOL

图 20-23　程序标签结构

四、测试应用

1. 将 SLogix5000 项目下载到控制器，方法为在主菜单中选择“通信→在线节点”（Communications→Who Active），然后选择位于机架 1 槽中的 ControlLogix 控制器，如图 20-24 所示。

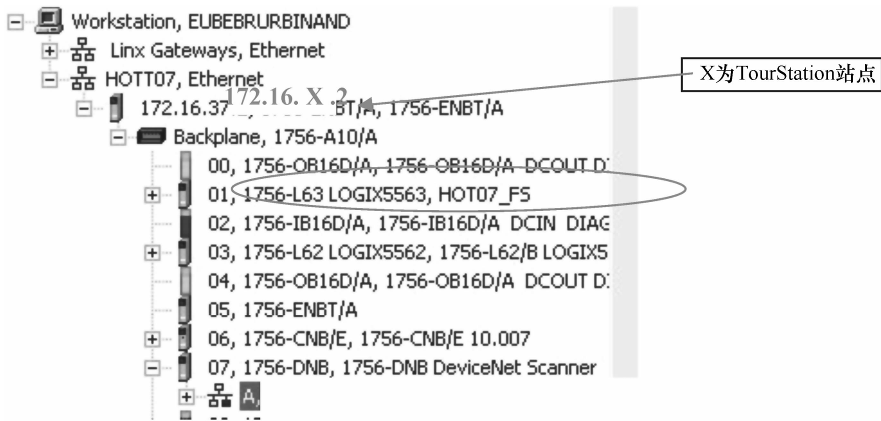


图 20-24　项目列表

2. 回答后续问题以下载项目，确保处理器处于运行模式，并且“I/O 正常”（I/O OK）指示为绿色常亮，如图 20-25 所示。

3. 请确保变频器没有运行并且无故障，控制 AOI 应如图 20-26 所示。



图 20-25　处理器工作模式

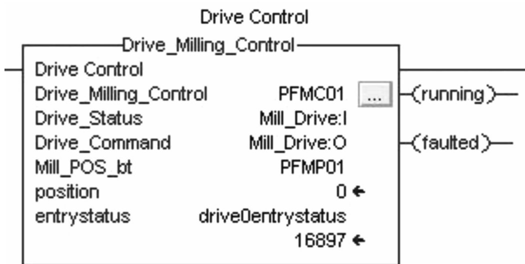


图 20-26　控制指令

4. 在工作站中的 Powerflex 70 上按下“ALT”，然后按“ +/-”按钮。将调出一个参数对话框。按“8”、“2”，然后“回车”，会看到参数#82 的值。请记住这个值。

5. 在控制指令上，将“位置”（position）的值设置为 1，如图 20-27 所示。

6. 验证“Drive_Milling_Position”指令是否成功执行，如图 20-28 所示。

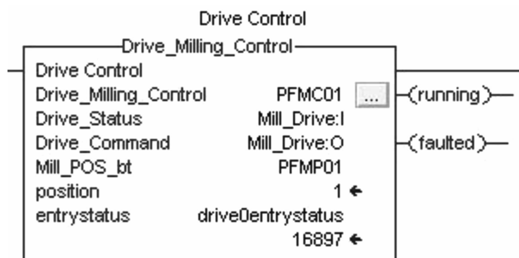


图 20-27 控制指令

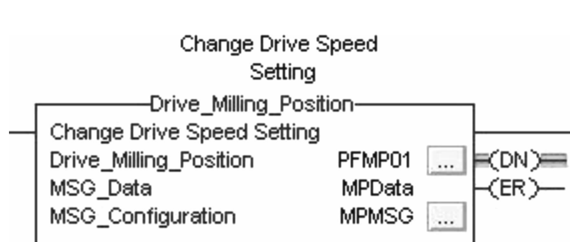


图 20-28 Drive_Millng_Position 指令

7. 在 Powerflex 显示界面中，观察参数 82 的值已经改变。

8. 该指令改变的参数：

55：最大频率

82：最大速度

83：超速限制值

140：加速时间 1

142：减速时间 1

190：方向模式

这些参数改变了 Powerflex 设备的动态特性，从而在定位模式和铣削模式之间切换。

9. 右键单击“PFMC01”并选择“Monitor“PFMC01””，如图 20-29 所示。

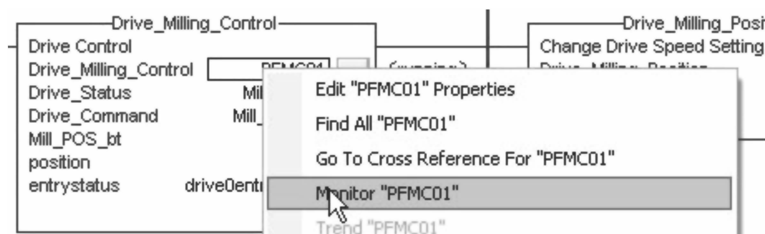


图 20-29 Monitor“PFMC01”命令

展开标签可以看到图 20-30。

10. 如果 Powerflex 存在故障，则将“PFMC01. reset”设置为 1。这样将复位变频器。该标签将立刻变回 0。

11. 请确保“PFMC01. position”为 1，并将“PFMC01. jogfwd”设置为 1。确保电动机在点动。

12. 再次将其设置为 0 并将“PFMC01. jogfrev”设置为 1。确保电动机在另一个方向点动。再设置为 0。

13. 将“PFMC01. position”设置为 0，然后切换“PFMC01. start”，方法是选择“PFMC01. start”并按 CTRL-T。将会看到电动机快速旋转。切换“PFMC01. stop”，看到电动机停止。

− PFMC01	{...}	Drive_Milling_Control
PFMC01.EnableIn	1	BOOL
PFMC01.EnableOut	0	BOOL
PFMC01.position	1	BOOL
PFMC01.start	0	BOOL
PFMC01.stop	0	BOOL
PFMC01.jogfwd	0	BOOL
PFMC01.jogrev	0	BOOL
PFMC01.reset	0	BOOL
+ PFMC01.entrystatus	16897	DINT
PFMC01.running	0	BOOL
PFMC01.faulted	0	BOOL

图 20-30　PFMC01 列表

五、查看逻辑和通过数据背景监视

1. 在项目管理器中打开主例程，右键单击“Drive_Milling_Control” AOI，选择“打开指令逻辑”（Open Instruction Logic），如图 20-31 所示。

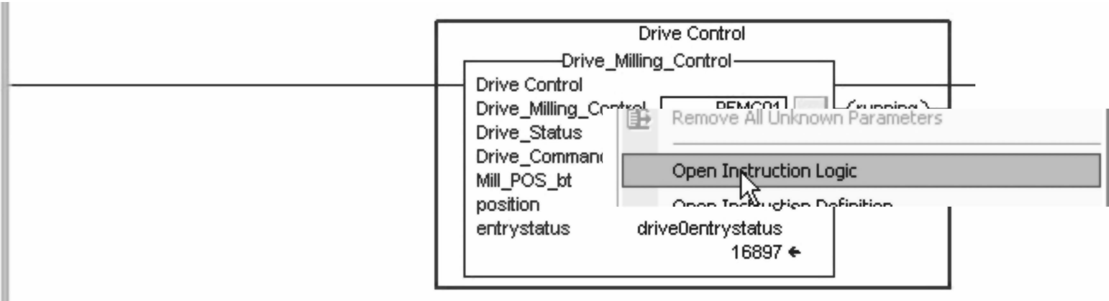


图 20-31　打开指令逻辑命令

2. 语言编辑器打开，包含用户自定义指令逻辑例程和通过指令调用的数据值。

3. 查看逻辑时，可以：

1）确定标签被用于数据的指令调用；

2）在逻辑执行时进行查看（离线）；

3）查看参数和局部标签值；

4）更改所选数据实例的局部标签和参数值。

4. 现在，请看语言编辑器的右上角。会看到这个工具栏，如图 20-32 所示。



图 20-32　工具栏

数据背景能够确定用于数据的指令调用（通过实例标签）。要编辑用户自定义指令的逻辑，必须在数据背景中选择指令 < 定义 >（ < definition > ），如图 20-33 所示。

确保数据背景与主程序对应。会在此处看到主程序中使用的 AOI 的特定实例（见图 20-34）。

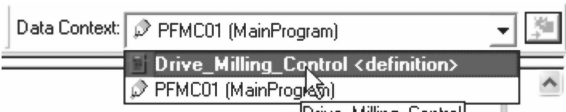


图 20-33　定义命令

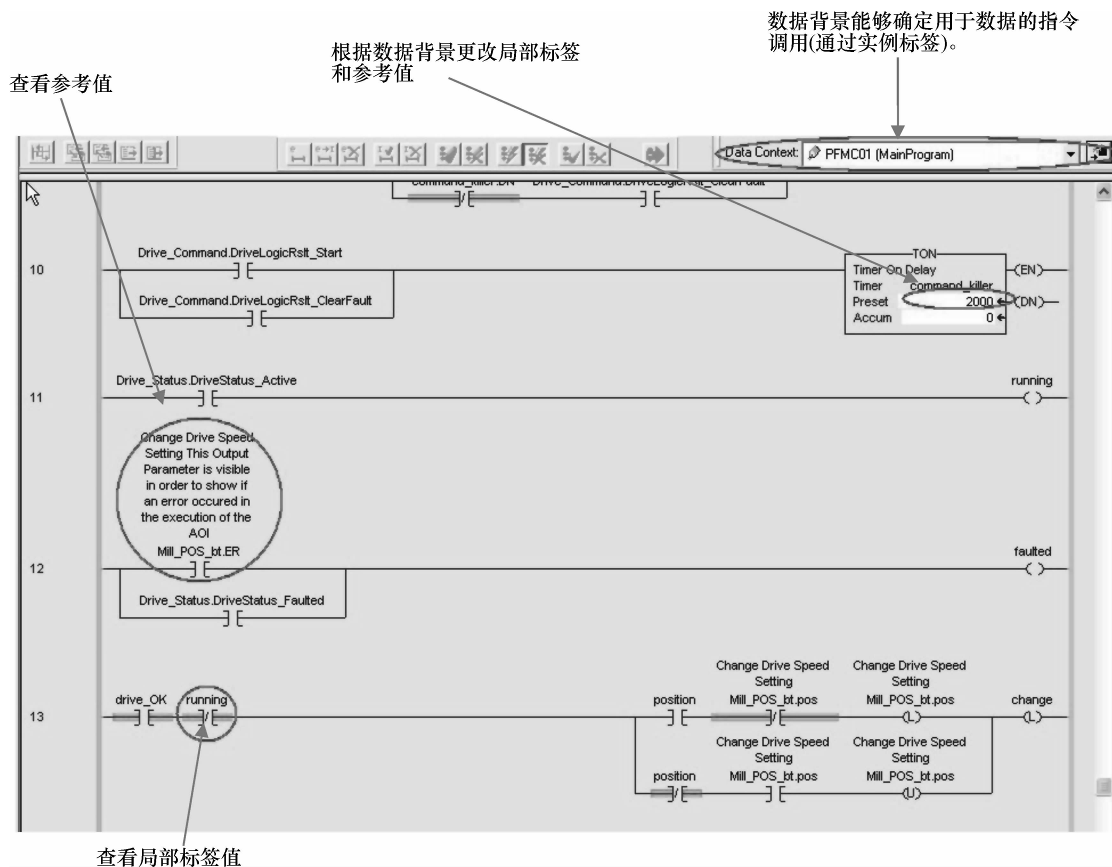


图 20-34 语言编辑器窗口

当在线时或逻辑在指令调用环境中时，或者逻辑受源代码保护时，都无法编辑逻辑。

5. 现在，请查看位于数据背景工具栏右侧的按钮，如图 20-35 所示。

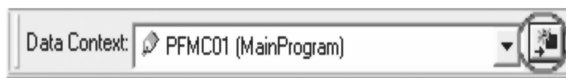


图 20-35 工具栏

6. 单击该按钮，将转到在数据背景中选择的 AOI 实例，如图 20-36 所示。

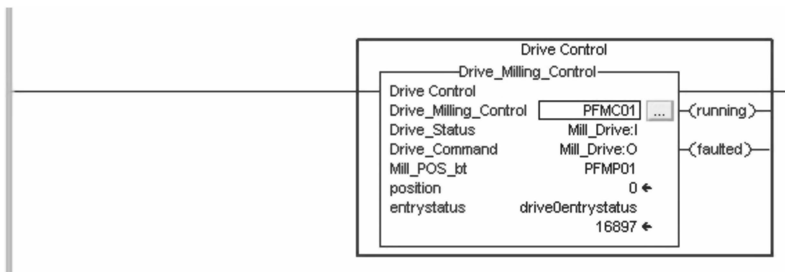


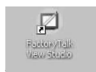
图 20-36 AOI 实例

六、基于 AOI，在 HMI 中使用全局对象

1. FactoryTalk View 的强大功能之一是能够创建全局对象。全局对象允许将图形对象的外观和行为链接到同一应用中该对象的多个副本。在更改原始对象时，副本也会随之变化。

在全局对象显示界面中创建全局对象。可在“全局对象”（Global Objects）文件夹中创建全局对象显示界面，这与在“显示界面”（Displays）文件夹创建标准显示界面的方法相同。全局对象显示界面中的所有对象和对象组都是全局对象。

2. 在桌面上双击以下图标打开 FactoryTalk View：



3. 在以下窗口中，选择“Machine Edition”并单击“继续”（Continue），如图 20-37 所示。

4. 在下一窗口中，选择“Milling_Machine”应用，如图 20-38 所示。



图 20-37　应用程序类型选择对话框

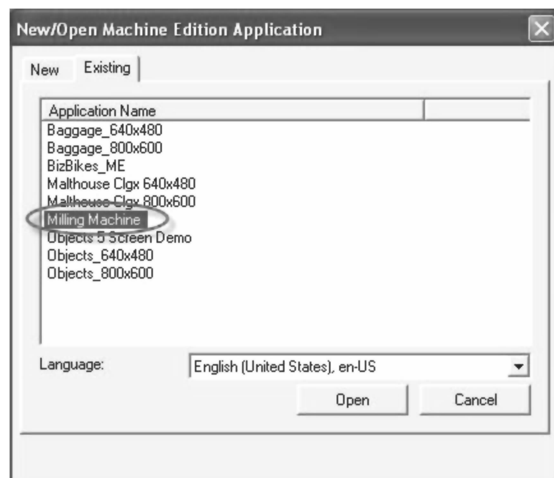


图 20-38　机器版本对话框

5. 在浏览器窗口中，打开“全局对象”（Global Objects）文件夹并双击“Milling_Object”，如图 20-39 所示。

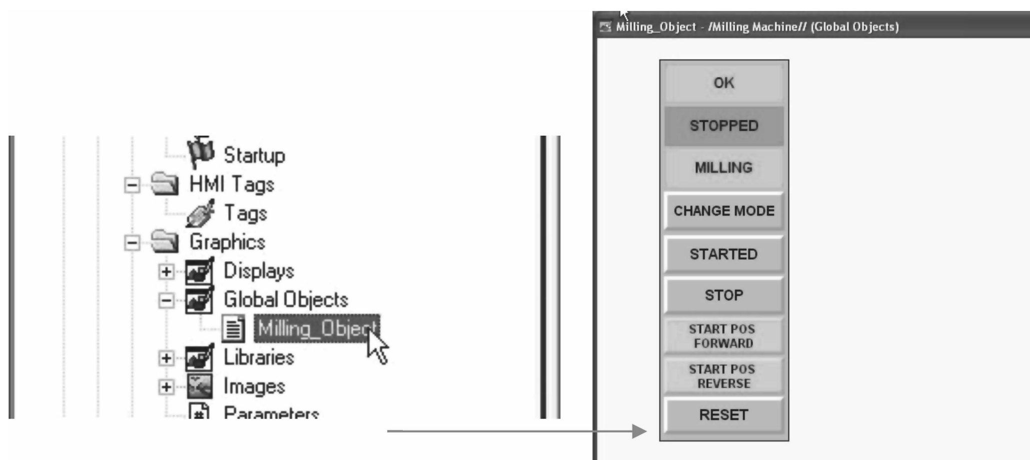


图 20-39　全局对象窗口

6. 右键单击全局对象并选择“复制”（Copy）。打开“显示界面”（Display）文件夹并双击“Milling_Display”，然后将该对象粘贴到空显示界面中，如图 20-40 所示。

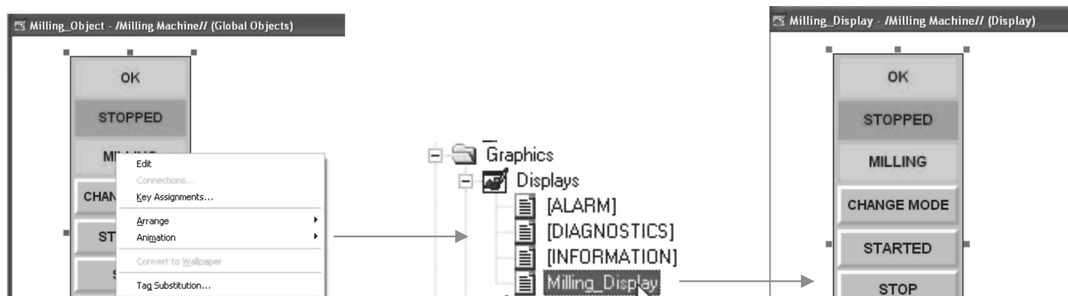


图 20-40　复制粘贴对象

7. 关闭“全局对象”（Global Object）窗口。右键单击位于新显示界面中的对象并选择“标签替换”（Tag Substitution），如图 20-41 所示。

8. 现在，可以使用全局对象了，并且该对象需要的所有数据都在“PFmc01” AOI 实例的支持标签中。只需替换一次，整个对象就会指向正确的标签成员。

在“搜索”（Search for）部分填入“#1”，在“替换为”（Replace with）部分填入“［Mill］Program：MainProgram. PFMC01”，截图如图 20-42 所示。

9. 单击“替换”（Replace）。在下一对话框中，单击“替换全部”（Replace All），如图 20-43 所示。

确保完成了 17 次替换，如图 20-44 所示。

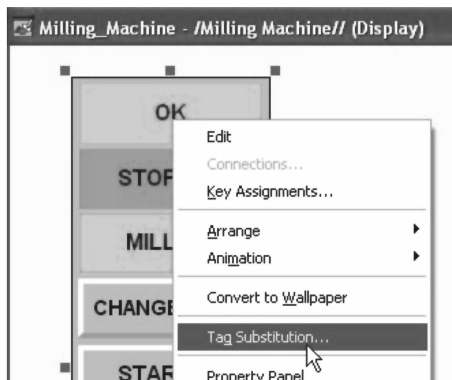


图 20-41　标签替换命令

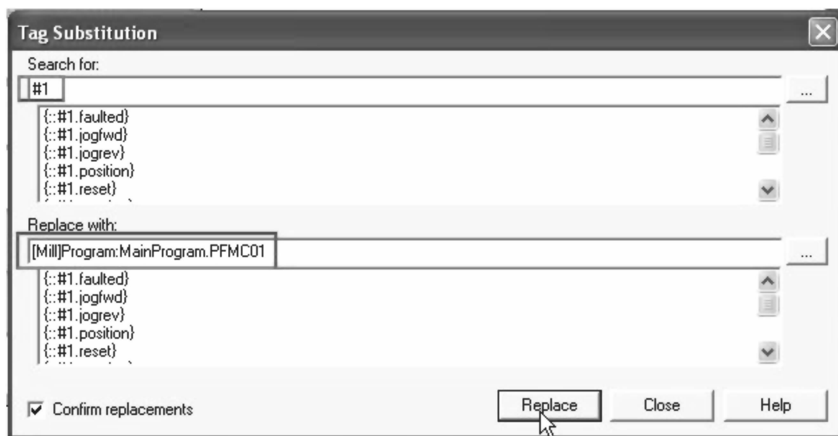


图 20-42　查找替换窗口

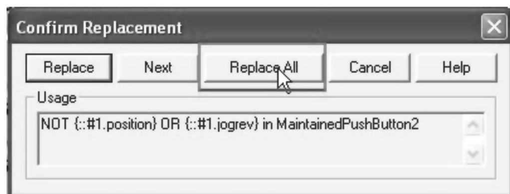


图 20-43　替换全部按钮

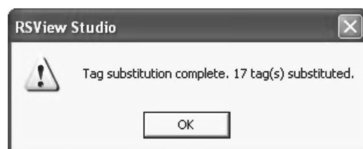


图 20-44　替换完成提示框

10. 现在，将模拟 View 应用，因此请按下“测试显示界面”（Test Display）按钮，如图 20-45 所示。可以测试铣床了。

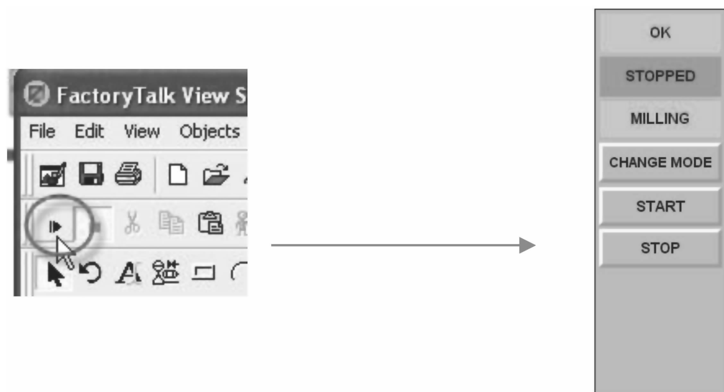


图 20-45 测试显示界面

实验 21：管理和更新用户自定义指令

一、实验目的

通过修改代码来改进应用。

二、存储用户自定义指令

1. 有两种方法可以将一组指令定义存储在一起。第一种方法是将用户自定义指令存储在项目文件中。为此，需要：

1）明确要存储哪些指令；

2）将它们置于某一项目文件中，名称可以是 MyInstructions. ACD；

3）在 RSLogix5000 软件的另一实例中打开其他项目，使用复制和粘贴或者拖放，将指令定义的副本从一个项目移动到另一项目，如图 21-1 所示。

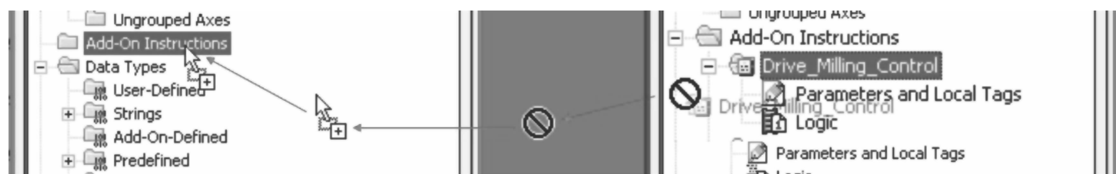


图 21-1 将指令定义的副本移动到另一个项目

如果这些指令定义引用了相同的用户自定义指令或用户自定义类型，则项目文件中只能存在一个共享副本。将用户自定义指令复制到另一项目时，还会将其引用的定义复制到目标项目中。

2. 第二种方法是将用户自定义指令存储在 . L5X 导出文件中。

选择导出用户自定义指令后，会将该用户自定义指令的定义保存到导出文件中。可以选择是否包含嵌套的用户自定义指令定义，或者所导出用户自定义指令中引用的用户自定义数据类型。在决定如何管理导出文件中的用户自定义指令定义时，需要考虑存储这些定义的目的。

是否希望将共用一套用户自定义指令或用户自定义类型的许多用户自定义指令存储到相同位置？

如果是，最好选择将各个用户自定义指令和用户自定义类型导出为单独的文件，而不包含引

用内容，步骤如下。

在控制器项目管理器中，选择“Drive_Milling_Control”用户自定义指令。右键单击该指令并选择“导出用户自定义指令”（Export Add-On Instruction），如图 21-2 所示。

选择存储 L5X 文件的位置，键入文件名，如图 21-3 所示。

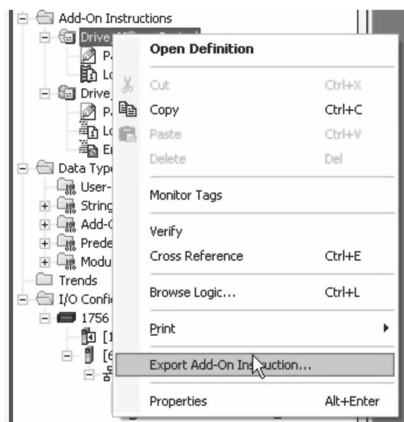


图 21-2 导出用户自定义指令

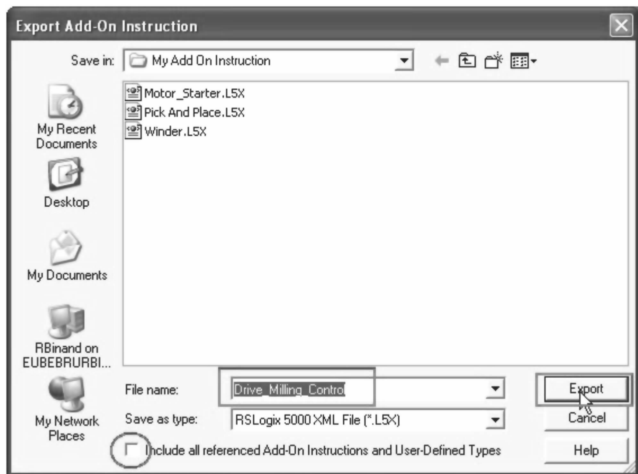


图 21-3 存储 L5X 文件

取消选中“包含引用的用户自定义指令和用户自定义类型”（Include referenced Add-On Instructions and User-Defined Types）复选框，然后单击“导出”（Export）。

按照以上步骤，将其他共用的用户自定义指令和用户自定义类型分别导出。

参考信息

按这种方法导出，可以将共用的用户自定义指令和用户自定义类型定义与引用它们的定义分开，单独进行管理。这种方法的优点是在更新共用部分时，不必为引用这部分的定义重新生成所有导出文件。也就是说，它仅存储在一个文件中，而不是存储在定义调用了该部分的所有文件中。这有助于定义的维护，因为仅需更新一个导出文件。

用这种方法导出用户自定义指令之后，要在项目中使用这些指令，必须先导入其引用的所有定义，而后才能成功导入引用这些定义的指令。为此，请按由下至上的方式，先导入嵌套的用户自定义指令或引用的用户自定义类型，然后是引用它们的用户自定义指令。

是否希望将用户自定义指令定义作为一个文件进行分配，或者将各个用户自定义指令作为独立的定义进行管理?

如果是，最好将该用户自定义指令及其引用的用户自定义指令或用户自定义类型导出到同一文件中，步骤如下。

在控制器项目管理器中选择“Drive_Milling_Control”用户自定义指令。右键单击该指令并选择“导出用户自定义指令”（Export Add-On Instruction），如图 21-4 所示。

选择 L5X 文件的存储位置。键入文件名，如图 21-5 所示。

选中“包含引用的用户自定义指令和用户自定义类型”（Include referenced Add-On Instructions and User-Defined Types）复选框，然后单击“导出”（Export）。

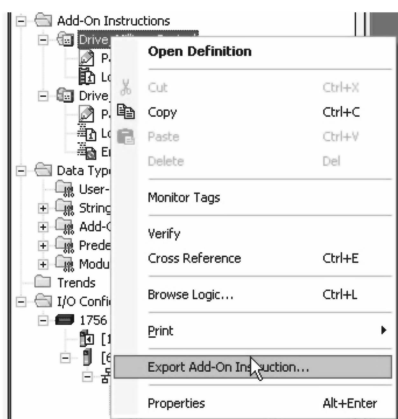


图 21-4　导出用户自定义指令

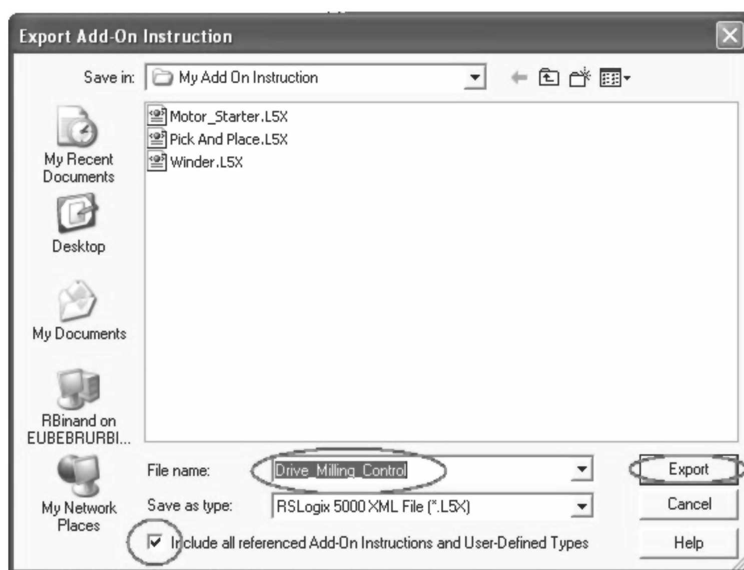


图 21-5　存储 L5X 文件

参考信息

这会将所选的用户自定义指令定义以及所有引用的定义导出到同一导出文件中。此文件可用于分配用户自定义指令定义。将导出的用户自定义指令导入到项目中时，此步骤还会将引用的定义一并导入。

三、将用户自定义指令更新到较新的版本

需要更新指令时，必须是在离线工作。如果需要较新版本的用户自定义指令，请从 L5X 文件导入或者从现有项目复制。请按以下步骤将用户自定义指令更新到较新的版本。

更新用户自定义指令时请谨慎操作。需要确保使用新版本的指令时现有代码仍能正常工作！

1. 在控制器项目管理器中选择“自定义指令”（Add-On Instruction）文件夹。右键单击该指令并选择“导入用户自定义指令”（Import Add-On Instruction）。

2. 从 C：\ HOTT \ AOI & Tools 文件夹中，选择文件“Drive_Milling_Position. L5X”，单击

“导入”（Import），如图 21-6 所示。

3. 检查“导入配置”（Import Configuration）对话框。因为 Drive_Milling_Control 引用了 Drive_Milling_Position，所以这里显示有两个定义，这两个定义都在导入文件中，如图 21-7 所示。

如果所导入的指令定义与项目中的定义存在冲突，则有两个选项适用于此用户自定义指令，“覆盖”（Overwrite）或“使用现有”（Use Existing）。

4. 由于现在知道如何备份 AOI，因此将用新的指令替换目前的 Drive_Milling_Control AOI。选择最后一行，然后单击“与现有比较”（Compare to Existing）按钮以查看“冲突”（Collision）对话框，如图 21-8 所示。

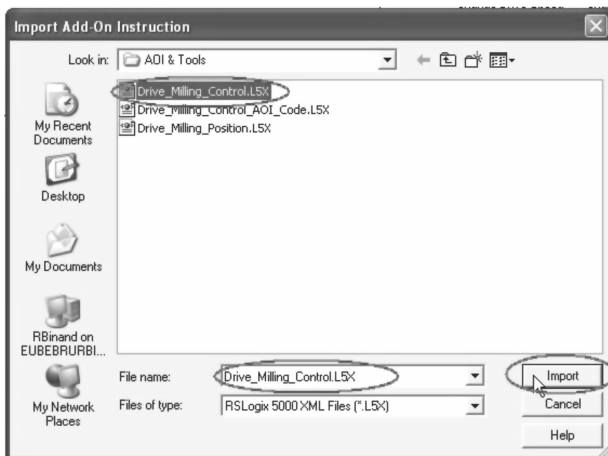


图 21-6 导出 L5X 文件

利用“冲突”（Collision）对话框查看两者区别以及该用户自定义指令的位置。

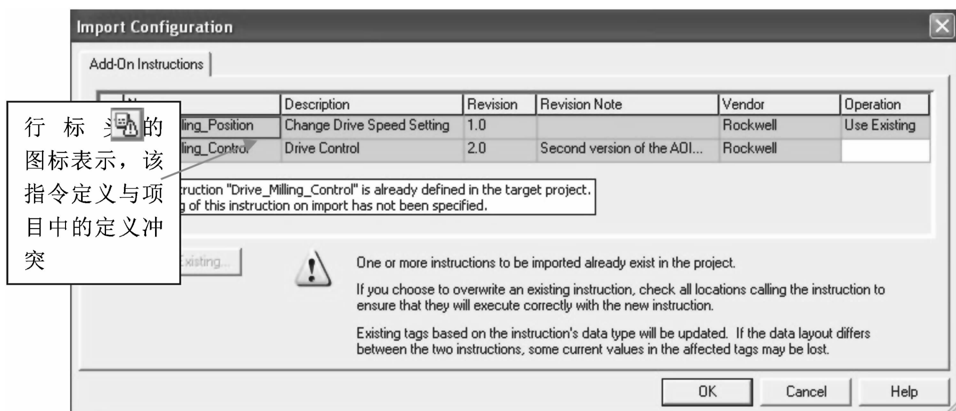


图 21-7 导入配置对话框

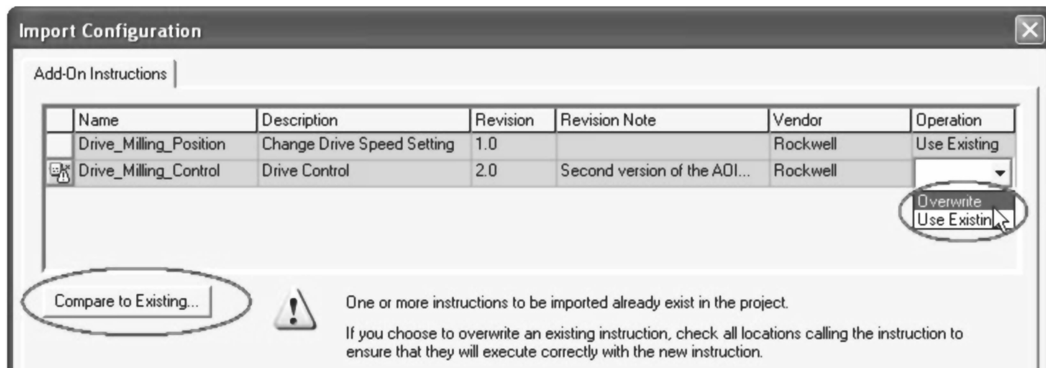


图 21-8 导入配置对话框

5. “冲突”（Collision）选项卡显示现有用户自定义指令的位置，如图 21-9 所示。

6. “比较”（Compare）选项卡显示这两个定义的区别，在本例中即版本信息，如图 21-10 所示。

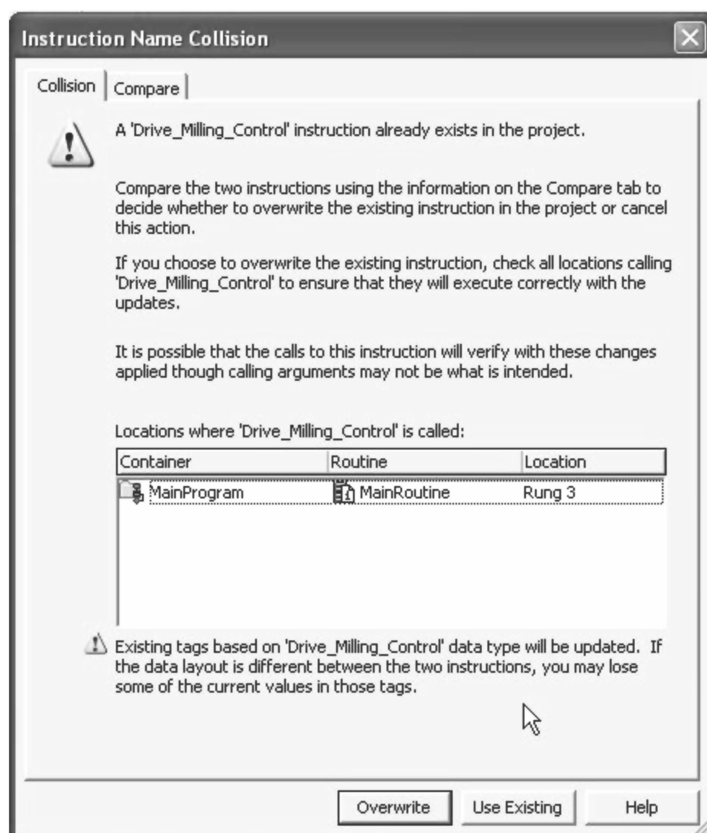


图 21-9　冲突对话框

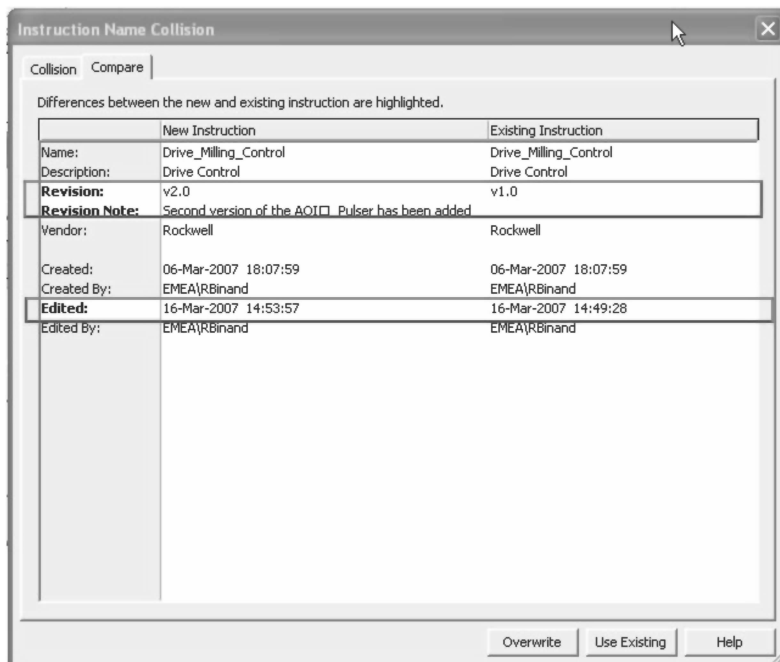


图 21-10　比较选项卡

7. 选择“覆盖”（Overwrite）用所导入的定义替换项目中的用户自定义指令。

8. 在“导入配置”（Import Configuration）对话框中单击“确定”（OK）完成此操作。将在错误窗口中看到，如图 21-11 所示信息。

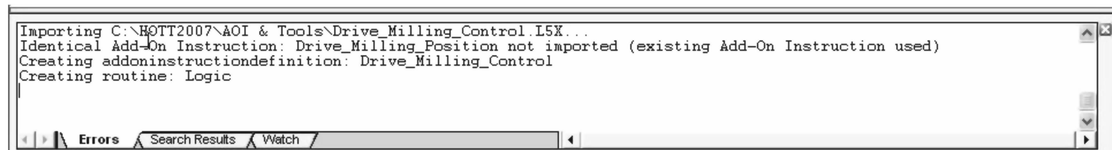


图 21-11 错误信息

实验 22：为 AOI 定义源代码保护

一、实验目的

在本实验中，将使用随 RSLogix 5000 光盘提供的免费源代码保护实用工具创建一个例程保护密钥文件。

参考信息

可通过应用源代码保护来保护的知识产权，或防止意外编辑已验证过的源代码。对用户自定义指令应用源代码保护能够保护的知识产权并避免意外的误修改。

可以使用“源代码保护”功能将使用用户自定义指令的某用户的访问权限设为只读，也可禁止用户访问该指令所使用的内部逻辑或本地标签。

在分配指令时，可以通过源代码密钥文件保护和控制对它们的使用和修改。这样可以防止意外更改的指令，也能够保护的知识产权。

有时，会希望在用户视图中保护用户自定义指令的源代码定义。这可能是由于算法的专有性，或者是出于严格版本控制的要求。

源代码保护

没有源代码密钥的用户无法查看例程或本地标签，也不能编辑用户自定义指令。

带可视选项的源代码保护

没有源代码密钥的用户可以查看用户自定义指令的所有部分，包括逻辑和本地标签，但不能编辑该指令。

从 RSLogix 5000 版本 8 开始，用户可以保护为 Logix 系列控制器编写的项目，使其免受未经授权的访问。RSLogix 5000 中的 OEM 保护在例程级别发挥作用。即，源代码受密钥保护；密钥对于项目中的各个例程可以是唯一的。

从 RSLogix 5000 版本 11 开始，源代码保护实用工具真正作为 RSLogix 5000 的一部分得到安装，因此用户可以在 RSLogix 5000 内部直接访问该实用工具，而不必通过单独的程序。

版本 11 的另一项新功能是可以选择将受保护的例程设为仅限查看，但无法编辑。在版本 11 之前，当某例程受到保护时是不提供仅限查看选项的。

二、保护 AOI 的内容

1. 在开始前，请确保已经离线，如图 22-1 所示。

2. 在 C：\ HOTT \ AOI & Tools 文件夹中，双击文件 RS5KSrcPtc. exe，启用 RSLogix5000 中的

源代码保护功能。

3. 将看到如图 22-2 所示窗口。

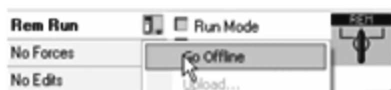


图 22-1　界面

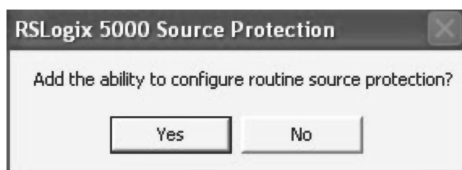


图 22-2　启用源代码保护对话框

4. 单击"是"（Yes）。在 RSLogix5000 项目中，单击"工具→安全→配置源代码保护"（Tools→Security→Configure Source Protection），如图 22-3 所示。

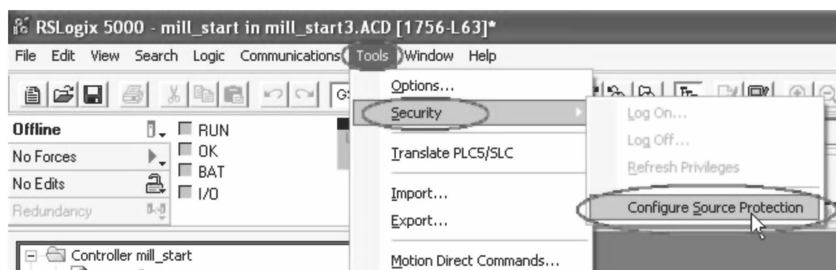


图 22-3　配置源代码保护对话框

5. 单击"是"（Yes）看到如图 22-4 提示。

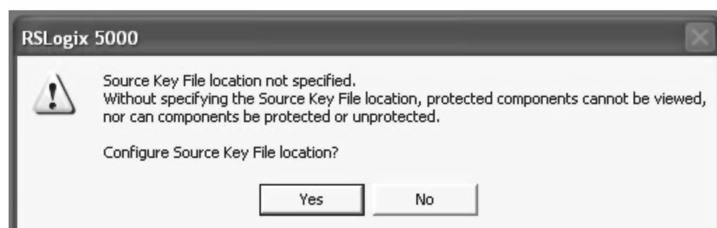


图 22-4　源代码保护对话框

6. 由于的计算机还没有源代码密钥文件，系统会提示选择创建该文件后的存储位置。单击 ... 省略号来为源代码密钥文件选择一个位置，如图 22-5 所示。

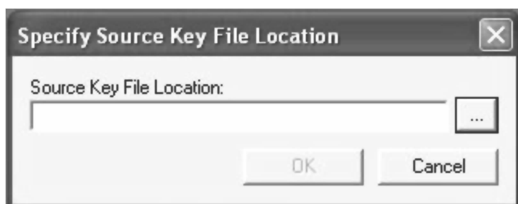


图 22-5　存储源代码对话框

7. 选择 C：\ HOTT \ AOI & Tools 文件夹，然后单击"确定"（OK），如图 22-6 所示。

8. 单击"确定"（OK）接受该路径，如图 22-7 所示。

9. 单击"是"（Yes）创建新的 sk. dat 文件，如图 22-8 所示。

10. 突出显示 Drive_Milling_Position 例程。会将此例程指定为秘密代码，不让任何人查看，如图 22-9 所示。

11. 单击"保护"（Protect）Protect... 按钮。

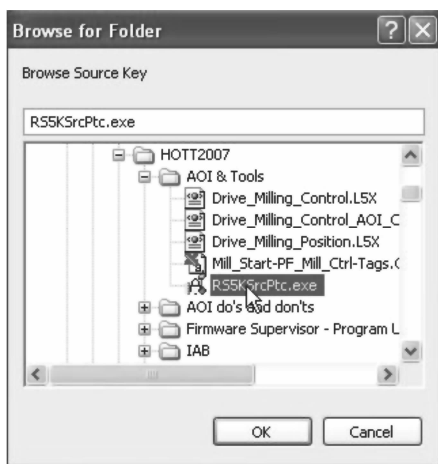


图 22-6 选择存储路径

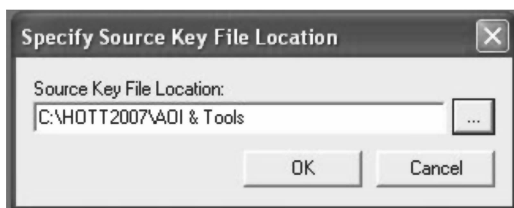


图 22-7 存储路径确认对话框



图 22-8 创建新的 sk. dat 文件

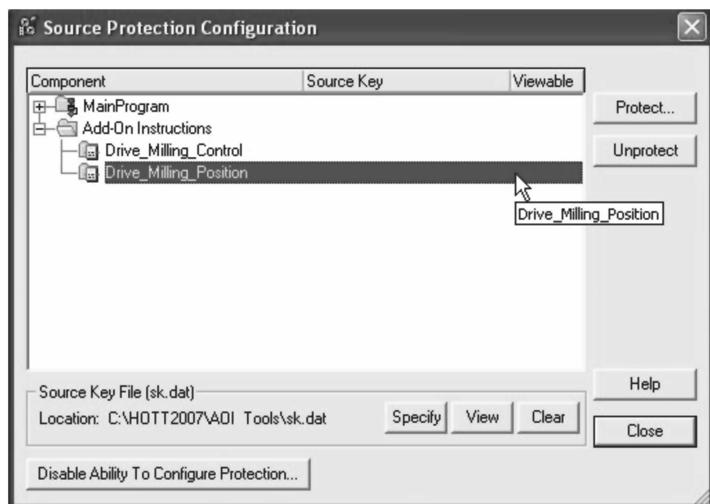


图 22-9 源代码保护配置对话框

12. 输入名称“secretaoi”作为源代码密钥的文件名，然后单击“确定”（OK），如图 22-10 所示。

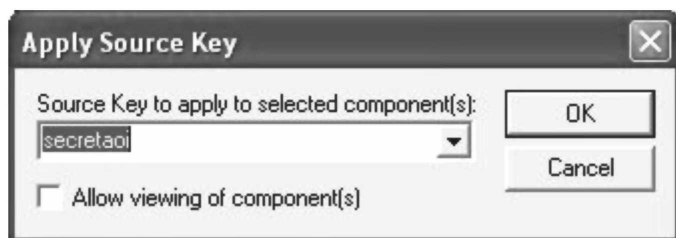


图 22-10 源代码密钥对话框

13. 突出显示 Drive_Milling_Control 例程。希望能够查看此例程，但不能进行编辑，如图 22-11 所示。

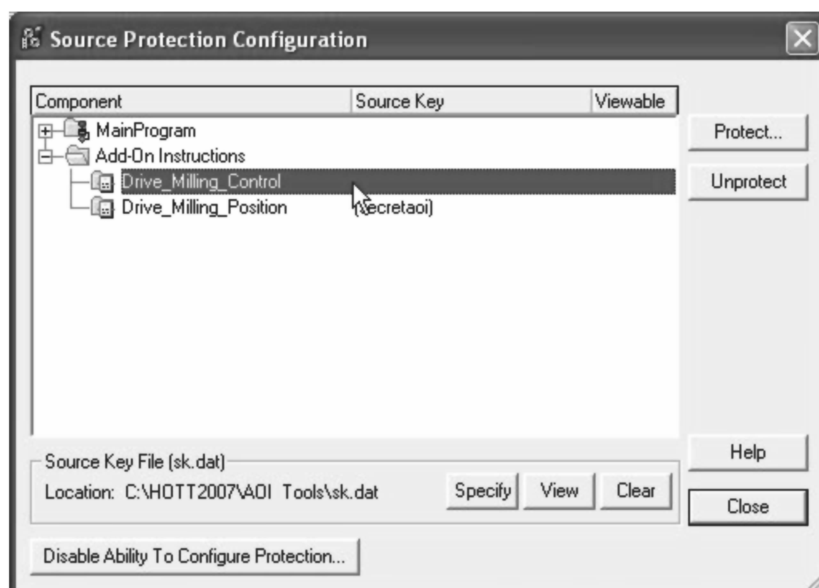


图 22-11　源代码保护配置对话框

14. 再次单击“保护”（Protect）按钮。

15. 将为此例程使用另一个源代码密钥，使得将来能够只取消其中一个例程的保护，同时保留对另一例程的保护。如果需要，是可以对两个例程使用同一源代码密钥的。键入源代码密钥的名称“view_only_aoi”，不包括引号。

16. 选中“允许查看组件”（Allow viewing of components）复选框，如图 22-12 所示。

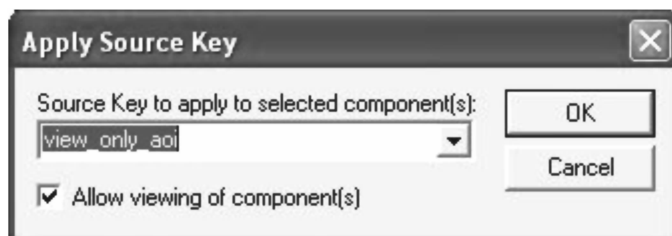


图 22-12　源代码密钥对话框

17. 单击“确定”（OK）。界面应如图 22-13 所示。

18. 关闭源代码保护配置窗口。

19. 将项目另存为“mill_code_protected. ACD”。

20. 关闭 RSLogix5000。

21. 打开 RSLogix 5000 mill_code_protected，然后确认可以查看这两个用户自定义指令的逻辑和指令定义。

在通过源代码密钥添加保护后，为何还能够查看这些例程呢？源代码密钥仍然在定义的位置，因此密钥都在它们对应的位置，拥有许可。一旦从指定位置移除源代码密钥，保护便会生效。

22. 完全关闭 RSLogix 5000，不是只关闭项目。

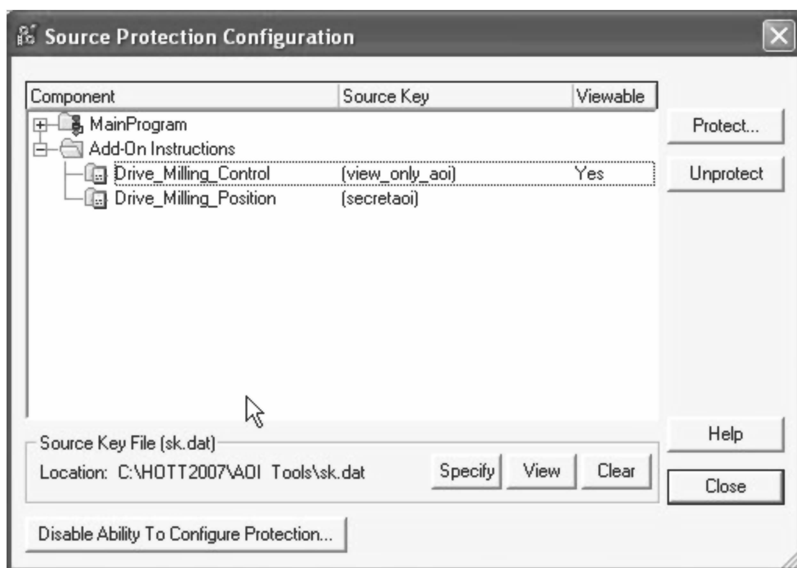


图 22-13 源代码保护配置对话框

23. 右键单击包含源代码密钥的文件，即 C：\ HOTT \ AOI & Tools 文件夹下的 sk. dat，如图 22-14 所示。

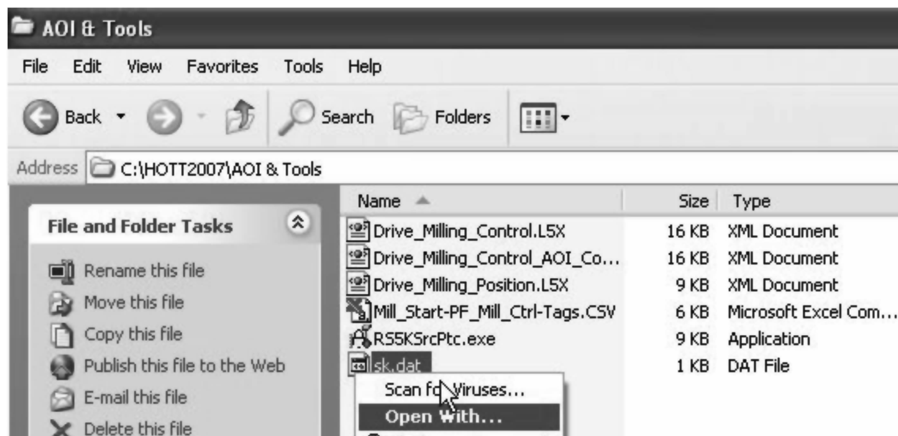


图 22-14 AOI & Tools 文件夹

24. 单击“打开方式…”（Open with…）。

25. 如果弹出以下警告，请单击“打开方式…”（Open With…）按钮，如图 22-15 所示。

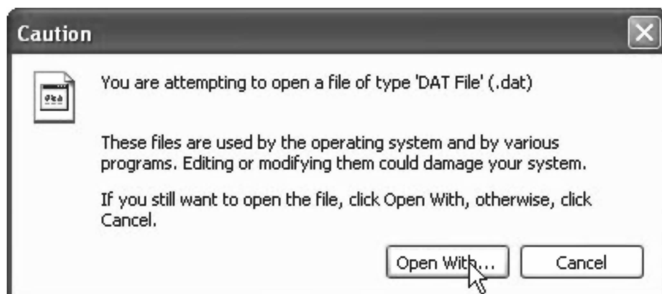


图 22-15 打开方式对话框

26. 选中“从列表中选择程序”（Select the program from a list）复选框，如图 22-16 所示。

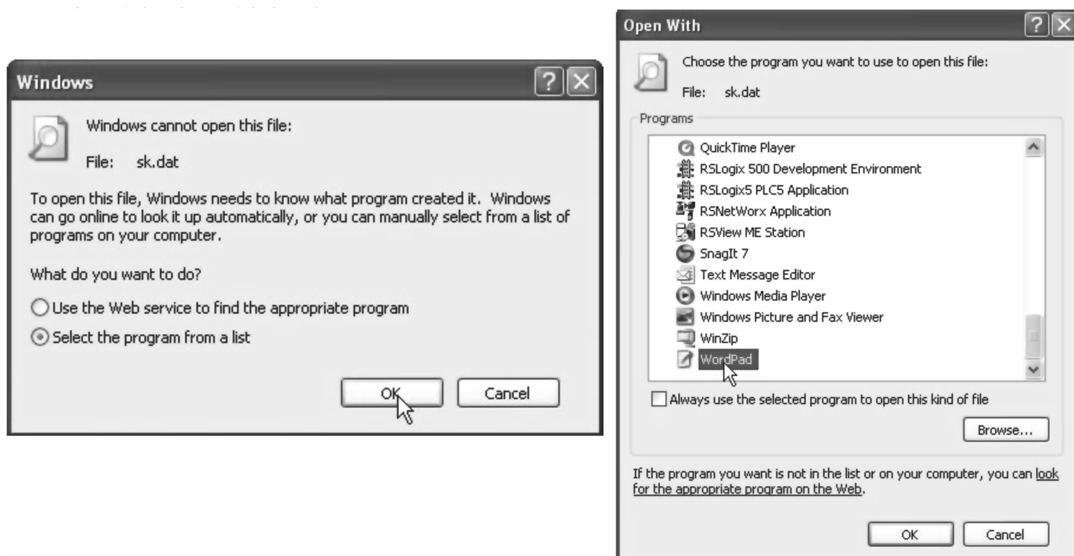


图 22-16　选择打开方式

27. 选择用写字板打开文件，单击“确定”（OK），将出现如图 22-17 所示界面。

因为源代码密钥在此位置，所以对于这些密钥在项目中对应的例程，拥有完全访问权。看一下这是如何实现的。

28. 从文件中删除“secretaoi”。

29. 关闭并保存该文件，单击“是”（Yes）看到如图 22-18 所示的提示。

30. 打开 RSLogix 5000。

31. 打开项目“mill_code_protected. ACD”。



图 22-17　写字板窗口

32. 请注意，Drive_Milling_Position 例程前已经没有“+”了，如图 22-19 所示。

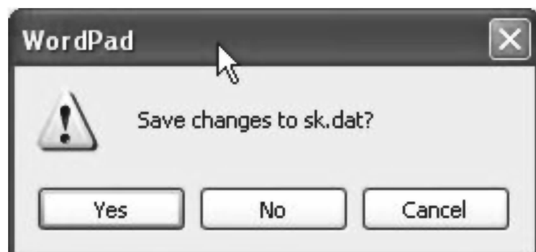


图 22-18　保存对话框

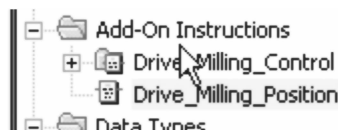


图 22-19　选择界面

33. 双击 Drive_Milling_Position。将打开指令定义对话框，但该对话框会灰显，并且左下角会提示“源代码不可用”（Source Not Available），如图 22-20 所示。

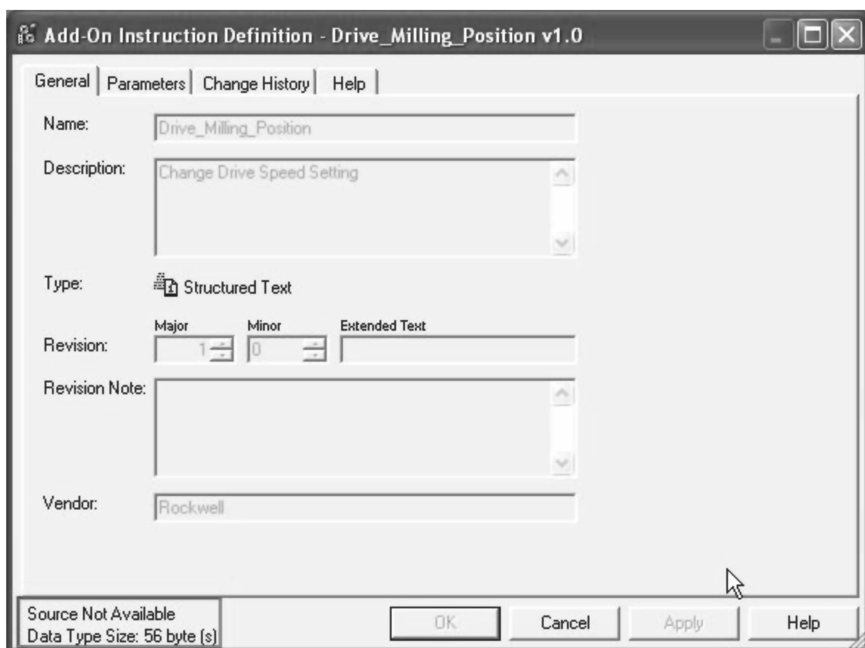


图 22-20　指令定义对话框

34. 双击 Drive_Milling_Control 自定义指令。会看到指令定义未灰显。

35. 双击 Drive_Milling_Control 逻辑。仍可以查看和访问该逻辑。

36. 关闭 RSLogix5000。

37. 回到 sk. dat 文件并删除 view_only_aoi。保存文件。

38. 再次打开 RSLogix 5000 mill_code_protected 项目。请注意，“ + ” 仍然在 Drive_Milling_Control AOI 前面，并没有像在保密的 Drive_Milling_Position AOI 中一样消失。

39. 双击 Drive_Milling_Control AOI。将打开指令定义对话框，不过是灰显的，如图 22-21 所示。

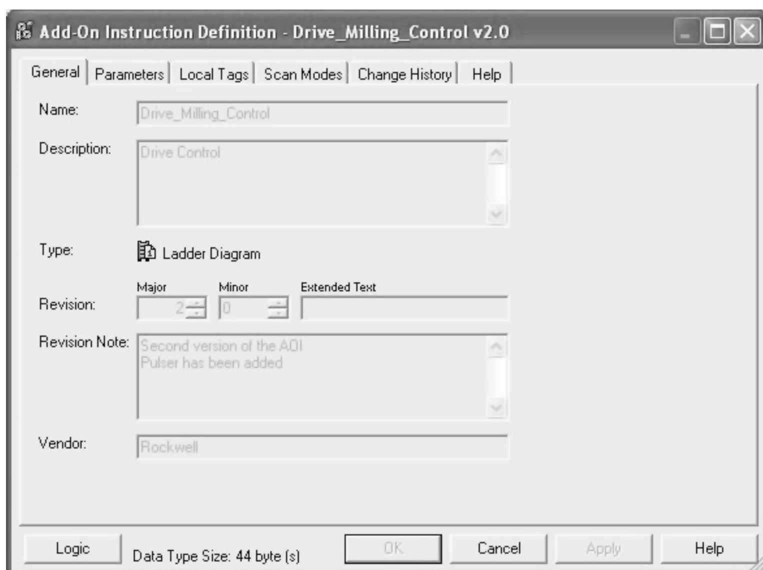


图 22-21　指令定义对话框

40. 双击 Drive_Milling_Control 逻辑。这次，还是能查看该逻辑，但没有权限进行更改，如图 22-22 所示。

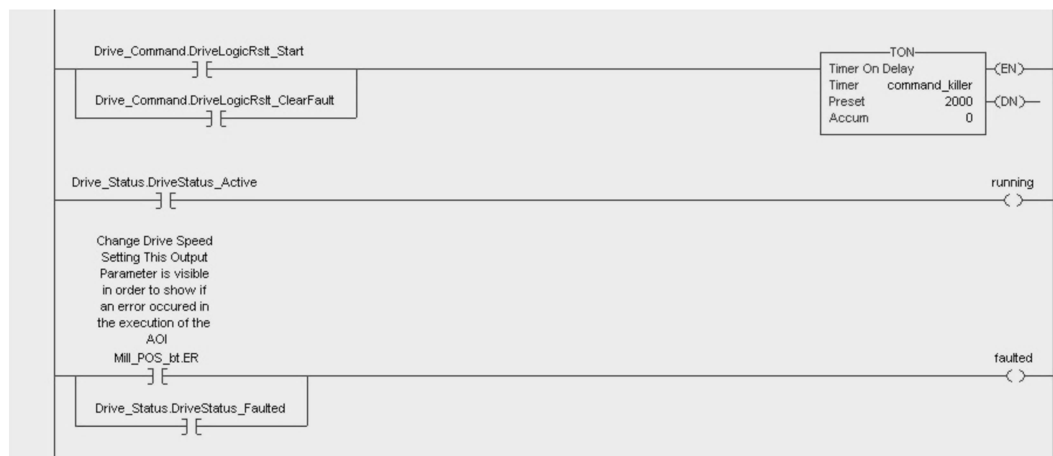


图 22-22　查看逻辑

实验 23：重用模块化对象

重用模块化对象

1. 现在，要将对象用于另一台 Powerflex 设备。这部分实验将是离线的。

2. 在 RSLogix 5000 项目中（离线），复制“mill_drive” Powerflex 并将其粘贴到以太网网络中，于是得到一个“mill_drive1”副本。将其重命名为 mill_drive2。

3. 在主例程中，复制并粘贴最后两个梯级。所复制代码的标签请使用：

1） Drive1entrystatus；

2） PFmc02；

3） Pfmp02；

4） mpMSG02；

5） mpdata02。

4. 必须按照之前的操作创建这些标签。确保 GSV 和 Drive_Milling_Control 指令均指向新变频器，像之前一样配置消息。

5. 在 FactoryTalk View 中，将全局对象再拖拽出一个实例，放到创建的显示界面中。

6. 通过标签替换，将“#1”替换为“［Mill］Program：MainProgram. PFMC02”。

第五部分　ME PowerFlex 面板

实验 24：ME PowerFlex 的使用

一、Logix 组态

1. 打开 RSLogix 5000 中现有的 Logix 文件。

2. 更新 PowerFlex 用户自定义配置文件。

PowerFlex 用户自定义配置文件与 RSLogix 5000 软件发布无关。更新 PowerFlex 用户自定义配置文件，确保应用运行无误。

可以在 RSLogix 5000 用户自定义配置文件网站上下载用户自定义配置文件，网址为 https：//download. rockwellautomation. com/esd/download. aspx？ downloadid = addonprofiles

请按照以下步骤下载 PowerFlex 用户自定义配置文件。

1）登录 RSLogix 5000 用户自定义配置文件网站，会打开 RSLogix 5000 用户自定义配置文件对话框，如图 24-1 所示。

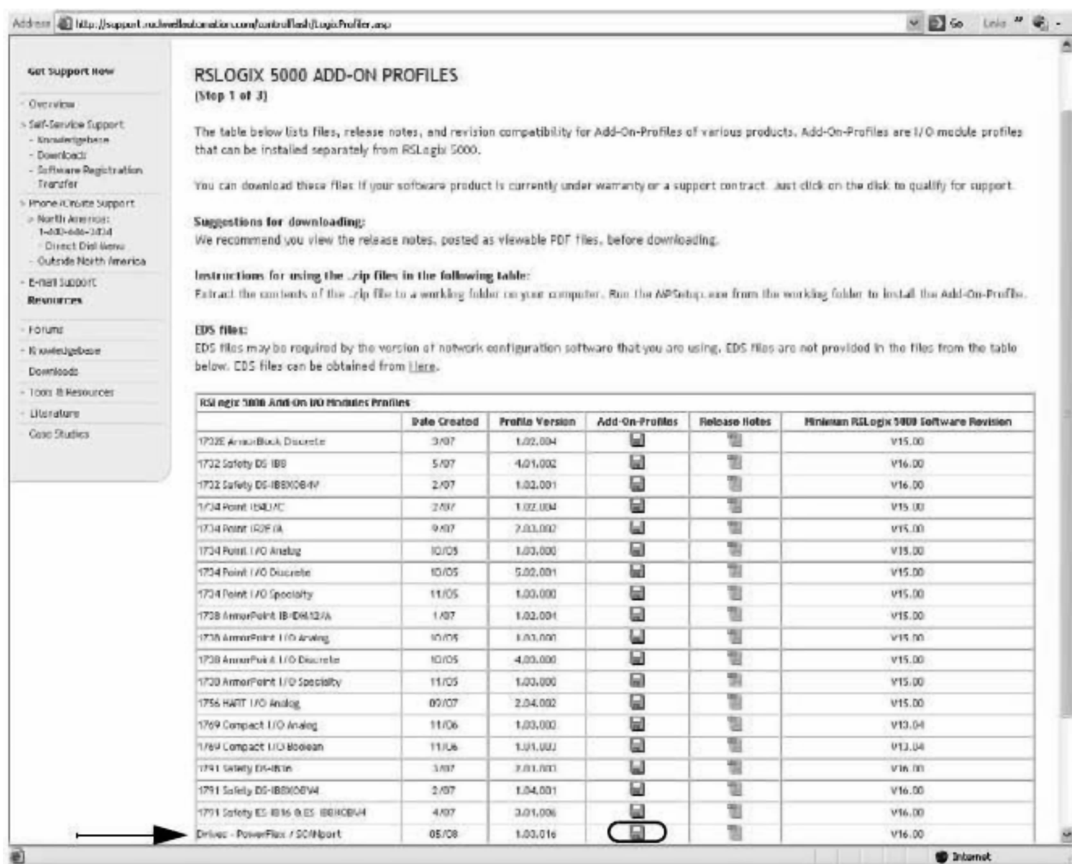


图 24-1　RSLogix 5000 用户自定义配置文件

2）找到 Drives-PowerFlex/SCANport 配置文件；

3）单击用户自定义配置文件图标并按照指令操作。

3. 添加并组态 PowerFlex 模块。

1）在“I/O 组态”（I/O Configuration）下，右键单击要配置的 PowerFlex 变频器所对应的网络通信模块，然后选择“新建模块”（New Module）。

2）打开“变频器”（Drives）文件夹，选择相应的 PowerFlex 变频器模块并选择“确定”（OK），如图 24-2 所示。

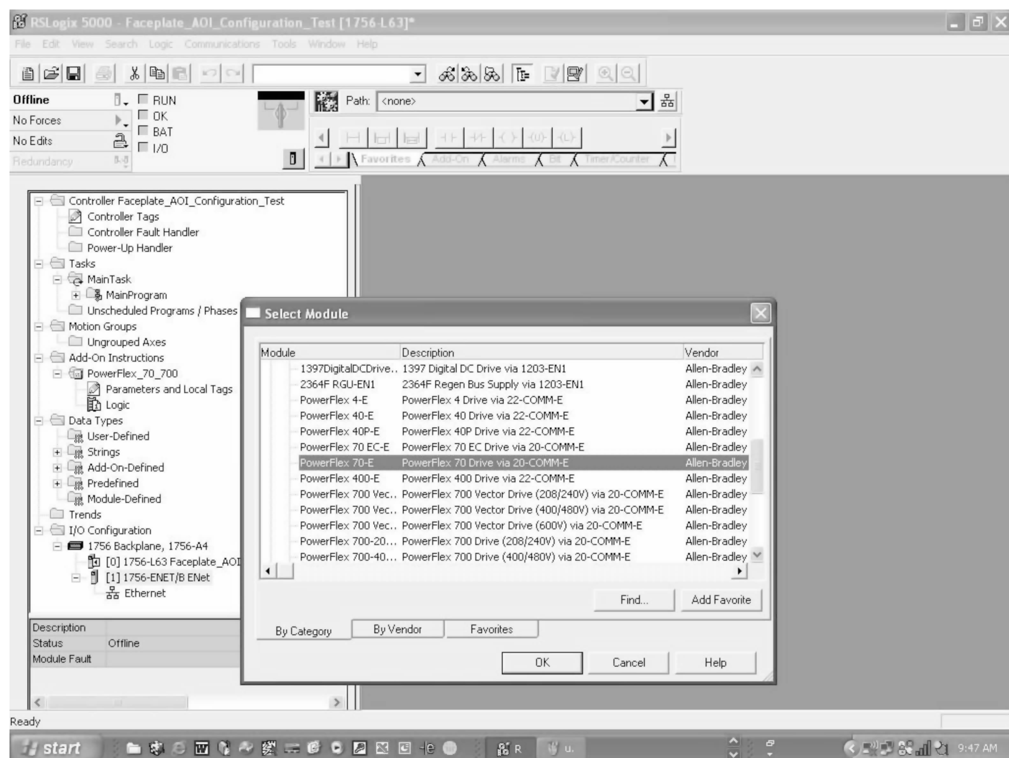


图 24-2　选择相应的变频器模块

3）输入不同的模块名称和 IP 地址，如图 24-3 所示。

4）选择“更改…”（Change…）按钮，然后为 PowerFlex 面板和 AOI 中使用的监视和诊断数据输入以下 * 默认数据链路模块定义。输入结束后单击“确定”（OK），如图 24-4、图 24-5 和图 24-6 所示。

* 注-PowerFlex 4、40 和 400 变频器不支持数据链路。因此，对应的 PowerFlex 4 面板不需要数据链路。

对于 PowerFlex 755 变频器，除了组态数据链路之外，还要验证是否选中“使用网络参考”（Use Network Reference）。

5）设置其他 PowerFlex 参数。

PowerFlex 参数设置特定于您的应用，并且应在运行变频器应用之前进行仔细检查。必须为常规的 Ethernet/IP 通信应用以下参数设置，从而使 PowerFlex 面板正常运行。

在 PowerFlex 模块属性的“变频器”（Drive）选项卡中，打开“参数列表”（Parameter List）并设置以下参数：

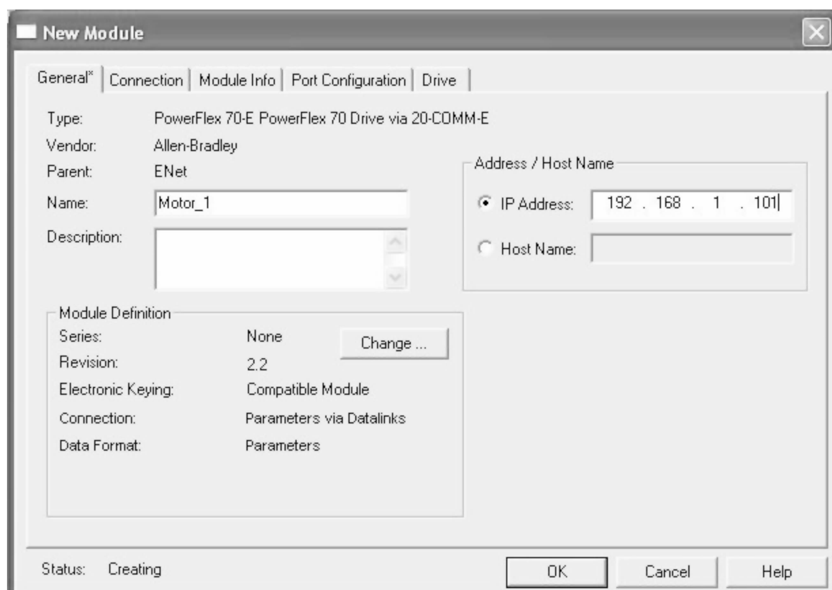


图 24-3　输入模块名称和 IP 地址

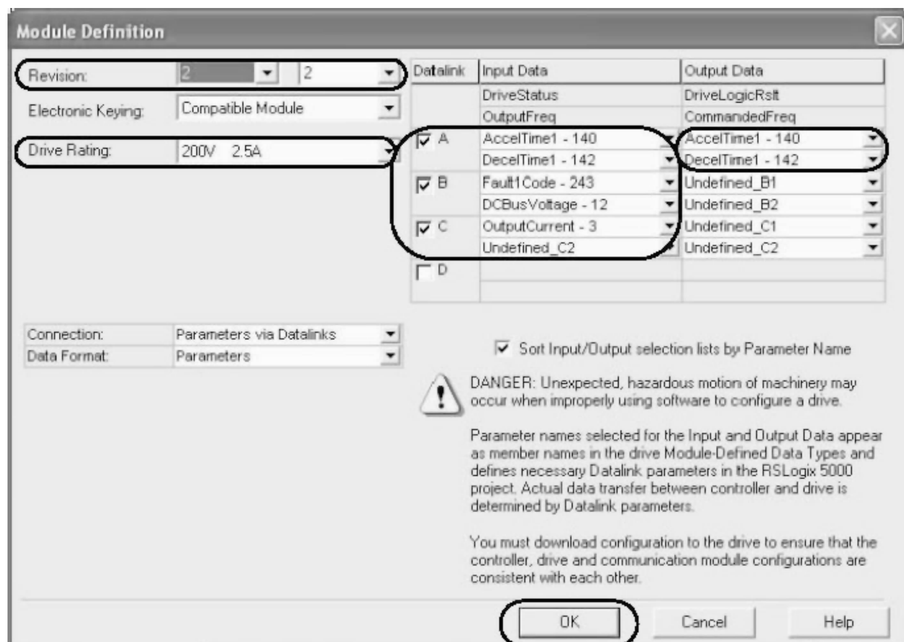


图 24-4　PowerFlex 70、70EC、700 和 700H 数据链路

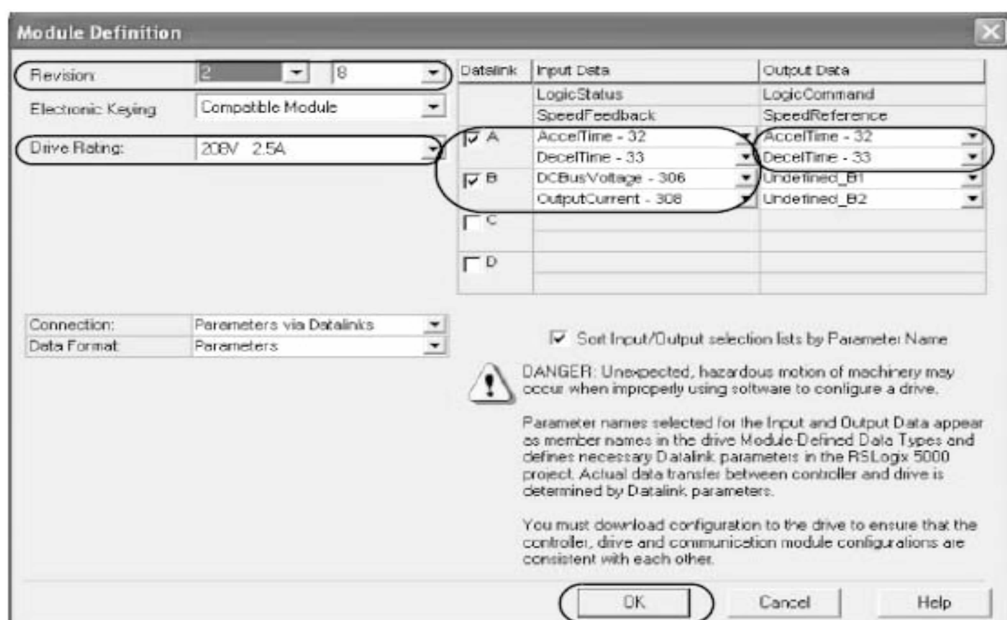


图 24-5 PowerFlex 700S 和 700S 2P 数据链路

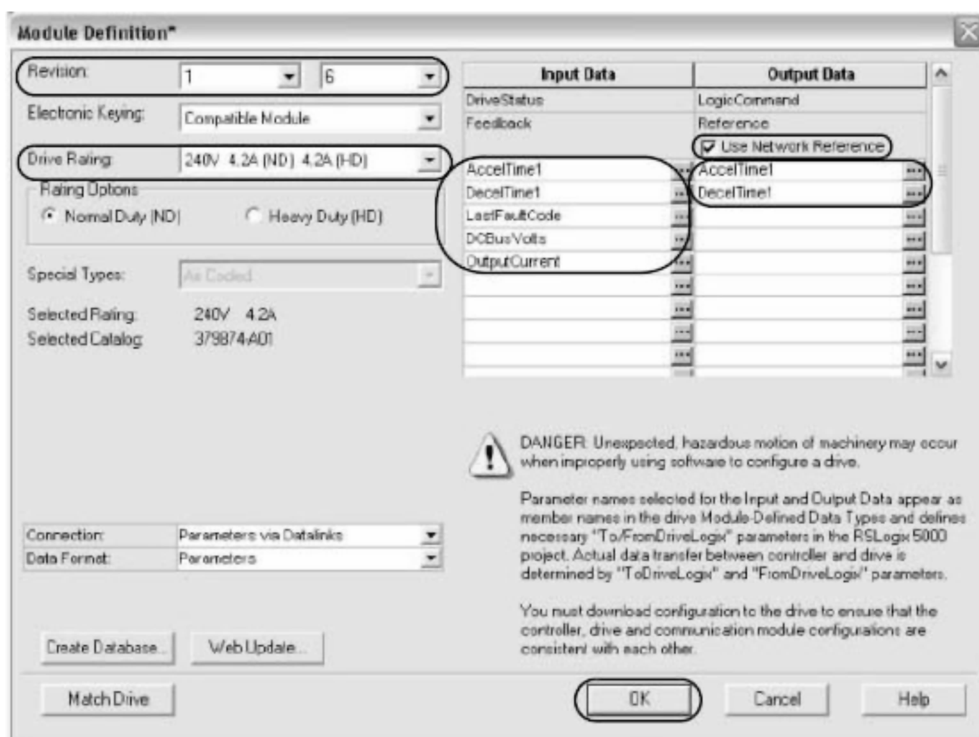


图 24-6 PowerFlex 755 数据链路

对于 PowerFlex 4、40、40P 和 400，将参数 36 和 38 设置为 Comm Port（5），如图 24-7 所示。

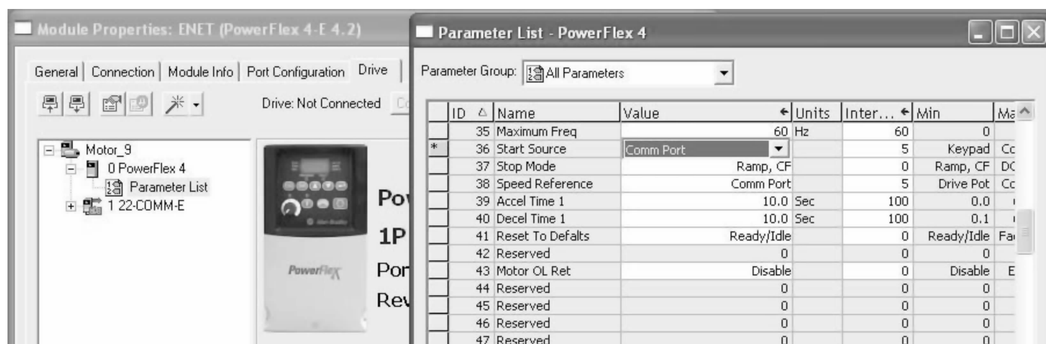


图 24-7　PowerFlex 4、40、40P 和 400 参数配置

对于 PowerFlex 70、70EC、700、700VC 和 700H，将参数 90 设置为 DPI Port 5（22），如图 24-8、图 24-9 所示。

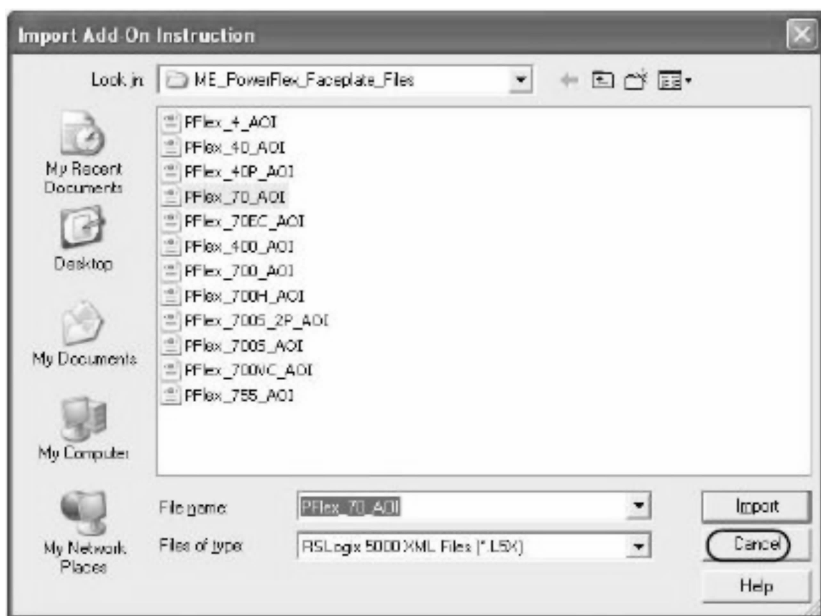


图 24-8　PowerFlex 70、70EC、700、700VC 和 700H

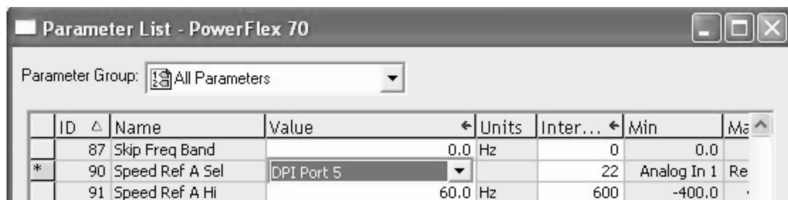
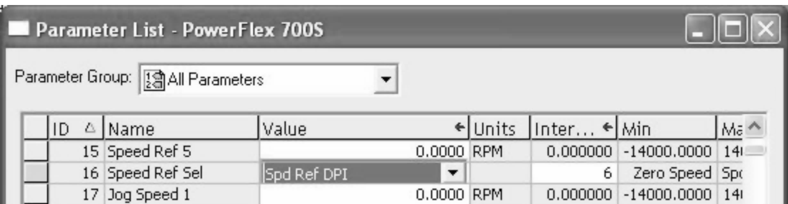


图 24-9　PowerFlex 70、70EC、700、700VC 和 700H 参数配置

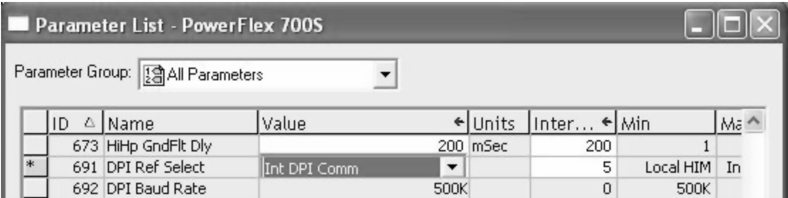
对于 PowerFlex 700S，将参数 16 设置为“Spd Ref DPI”（6），将参数 691 设置为“Int DPI Comm”（5），如图 24-10、图 24-11 所示。

对于 PowerFlex 700S 2P，将参数 27 设置为 DPI Port 5（16），如图 24-12 所示。



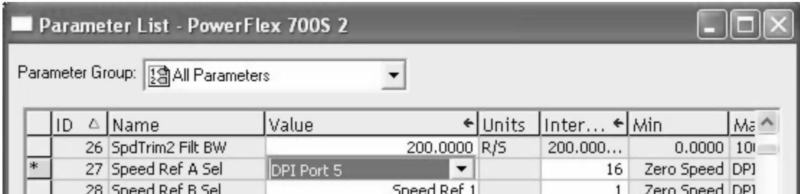
	ID	Name	Value	Units	Inter...	Min	Ma
	15	Speed Ref 5	0.0000	RPM	0.000000	-14000.0000	14
	16	Speed Ref Sel	Spd Ref DPI		6	Zero Speed	Sp
	17	Jog Speed 1	0.0000	RPM	0.000000	-14000.0000	14

图 24-10　PowerFlex 700S 参数配置 1



	ID	Name	Value	Units	Inter...	Min	Ma
	673	HiHp GndFlt Dly	200	mSec	200	1	
*	691	DPI Ref Select	Int DPI Comm		5	Local HIM	In
	692	DPI Baud Rate	500K		0	500K	

图 24-11　PowerFlex 700S 参数配置 2



	ID	Name	Value	Units	Inter...	Min	Ma
	26	SpdTrim2 Filt BW	200.0000	R/S	200.000...	0.0000	10
*	27	Speed Ref A Sel	DPI Port 5		16	Zero Speed	DPI
	28	Speed Ref B Sel	Speed Ref 1		1	Zero Speed	DPI

图 24-12　PowerFlex 700S 2P 参数配置

当选中“模块定义”（Module Definition）对话框中的“使用网络参考”（Use Network Reference）时，PowerFlex 755 的变频器速度给定值已经自动设定了。

4. 为您所使用的变频器导入 PowerFlex AOI 文件（即 PFlex_70_AOI 文件）。

1）在系统树中，右键单击“用户自定义指令”（Add-On Instructions），然后选择“导入用户自定义指令”（Import Add_On Instruction）。会出现“导入用户自定义指令”（Import Add-On Instruction）对话框。

2）浏览至“ME_PowerFlex_Faceplate 文件”（ME_PowerFlex_Faceplate Files）文件夹，并正在应用的变频器选择 AOI。

3）单击“导入”（Import）按钮，完成导入操作，如图 24-13 所示。

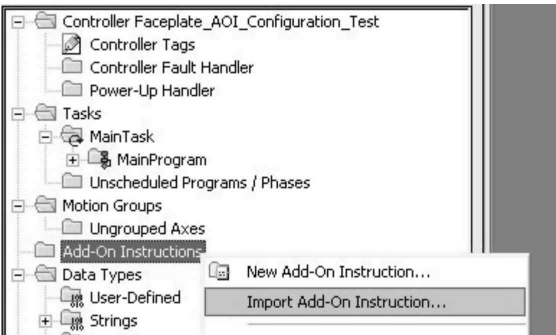


图 24-13　导入操作

注：如果对同种类型的变频器（如 PF 70）使用不同数据链路定义创建了多个 PowerFlex 模块定义，你也要为这些定义创建不同的 AOI，并使用不同的名字。

5. 重新为已组态变频器分配 AOI 中的“输入和输出”参数数据类型。

1）在项目文件 Explorer 窗口中，展开“用户自定义指令”（Add-On Instructions）文件夹，如图 24-14 所示。

2）选择您所使用的相应 PowerFlex AOI 文件，右键单击“参数和本地标签”（Parameters and

Local Tags）。本示例中选择的是 PFlex_70_AOI。

3）滚动列表，选择 Inp_PowerFlex_xx AOI 标签，并单击数据类型字段旁的 ⋯。将打开“选择数据类型”（Select Data Type）对话框，如图 24-15 所示。

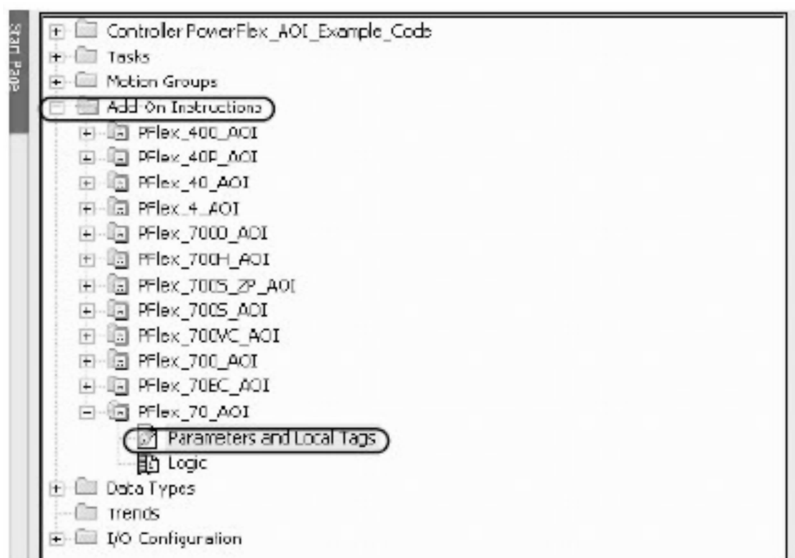


图 24-14 用户自定义指令文件夹

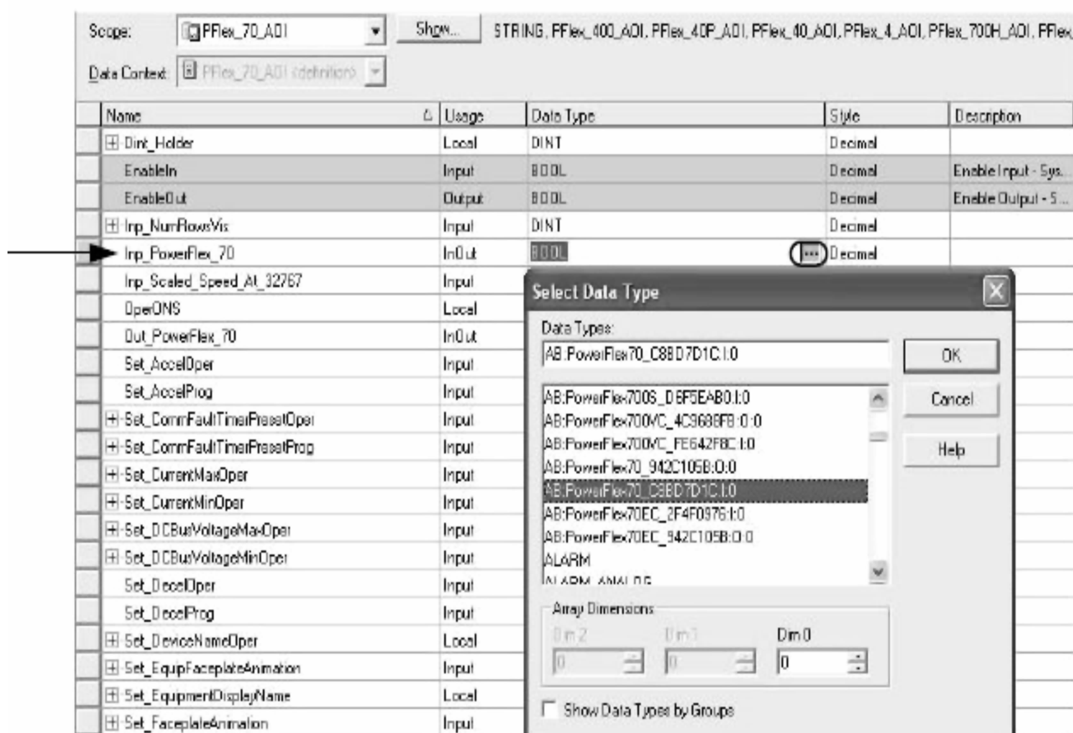


图 24-15 “选择数据类型”对话框

4）浏览至与您在步骤 3 中组态变频器时，所创建的 PowerFlex 模块数据类型相匹配的输入数

据类型。

其名称应该与所组态的变频器名称相似，如 AB：PowerFlex70_xxxxxxx：I：x。

重要说明：

如果对 PowerFlex 模块定义进行了多次更改并且保存了这些更改，则会为特定类型的变频器列出多种数据类型，因此将在 AOI 中选择的数据类型与预定的 PowerFlex 模块定义的数据类型匹配时要多加小心。

5）单击“确定”（OK）。

会更新“数据类型”（Data Type）字段。本示例显示的是 PowerFlex 70 输入标签的数据类型，如图 24-16 所示。

图 24-16　输入标签的数据类型

6）选择 Out_PowerFlex_xx AOI 标签并单击数据类型字段旁的…。将打开选择“数据类型”（Select Data Type）对话框，如图 24-17 所示。

图 24-17　打开“数据类型”对话框

7）浏览至与您在步骤 3 中组态变频器时所创建的 PowerFlex 模块数据类型相匹配的输出数据类型。

8）其名称应该与所组态的变频器名称相似，如 AB：PowerFlex70_xxxxxxx：O：x。

9）单击“确定”（OK）。

会更新“数据类型”（Data Type）字段。本示例显示的是 PowerFlex 70 输出标签的数据类型，如图 24-18 所示。

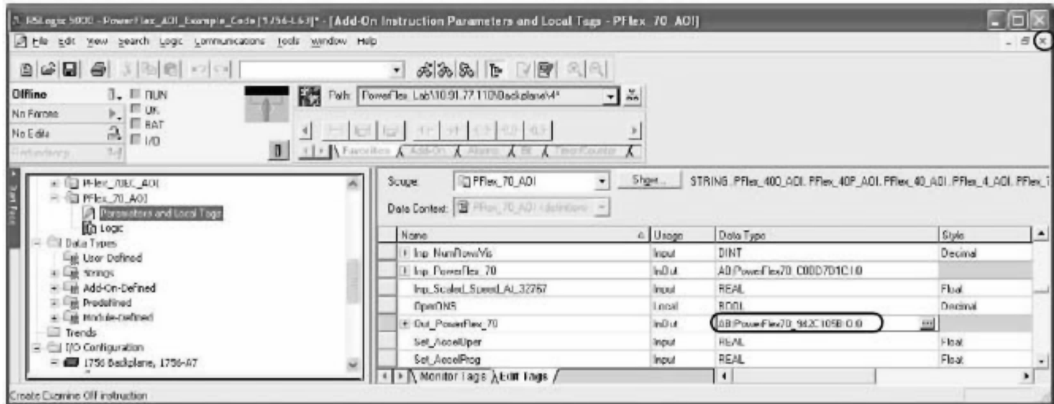


图 24-18 输出标签的数据类型

10）完成之后关闭“参数和本地标签”（Parameter and Local Tags）。

6. 将 PowerFlex AOIs 放到应用程序例程中。

1）打开预定的程序例程，创建梯级，然后选择“用户自定义指令”（Add-On instruction）选项卡。

2）选择 PowerFlex 用户自定义指令（即 PFlex 70），如图 24-19 所示。

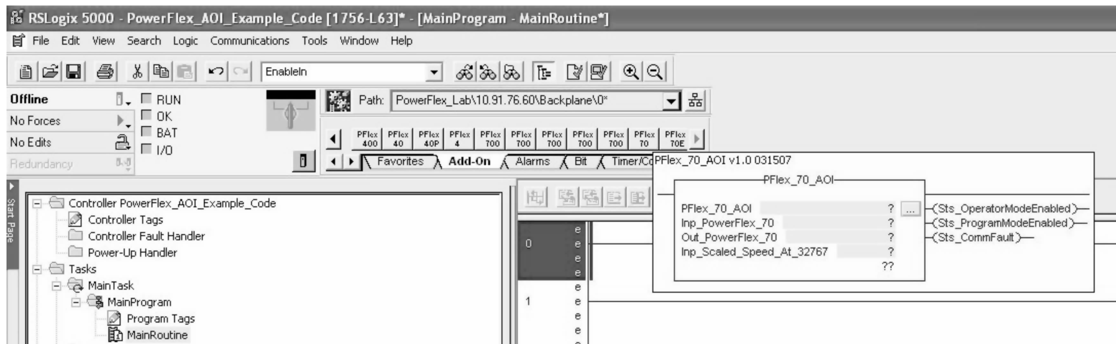


图 24-19 PowerFlex 用户自定义指令

3）将 PowerFlex 数据类型标签分配给与要控制和监视的模块名称相同的主 AOI 标签（本示例中为“Motor_1”）。然后，右键单击该标签，添加数据类型为“PFlex_70_AOI”的 Motor_1 标签，如图 24-20 所示。

4）为先前组态的 AOI 选择模块输入和输出标签。确保在添加标签之后保存应用文件，如图 24-21 所示。

5）输入 Inp_Scaled_Speed_At_xxxx 值，AOI 将根据所输入值的比例决定变频器的输入和输出

速度值，如图 24-22 所示。

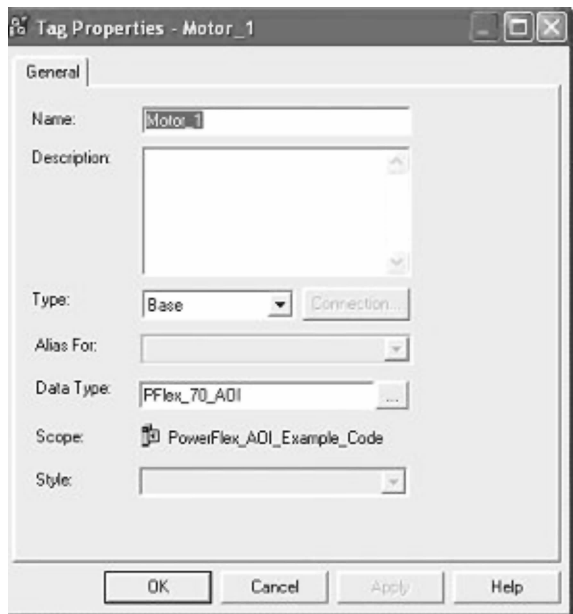


图 24-20　添加数据类型为“PFlex_70_AOI”的 Motor_1 标签

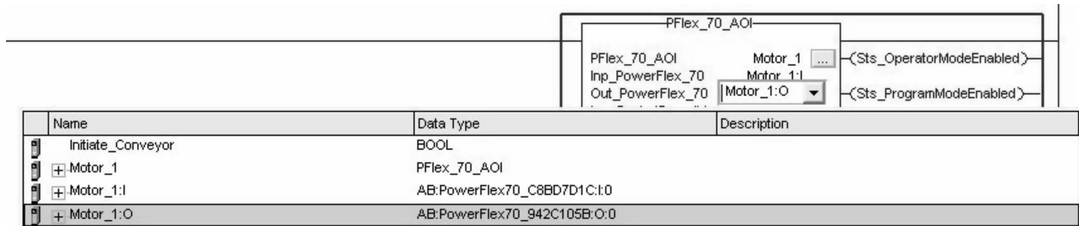


图 24-21　选择模块输入和输出标签

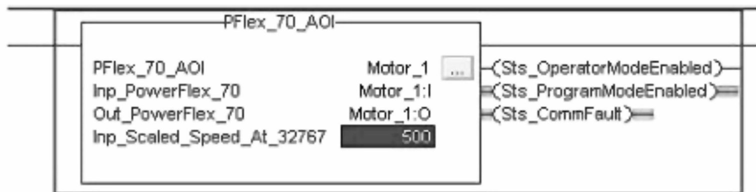


图 24-22　输入 Inp_Scaled_Speed_At_xxxx 值

7. 与特定 Logix 应用程序逻辑集成。

1）查看用于集成到特定 Logix 应用程序中的用户自定义标签和示例 Logix 代码。

a）展开“系统”（System）树中的“用户自定义”（Add-On Definition）文件夹，然后展开“PowerFlex 变频器 AOI”（PowerFlex Drive AOI）。可以查看要集成到应用程序中并对 PowerFlex 变频器进行程序控制的“参数和本地标签”（Parameters and Local Tags），如图 24-23 所示。

b）为了将 PowerFlex AOI 集成到特定 Logix 应用程序中一般应用进行示例，请打开“ME_PowerFlex_Faceplate 文件”（ME_PowerFlex_Faceplate Files）文件夹中的 PowerFlex_AOI_Example_Code Logix 应用文件 56，打开 MainRoutine 项，如图 24-24 所示。

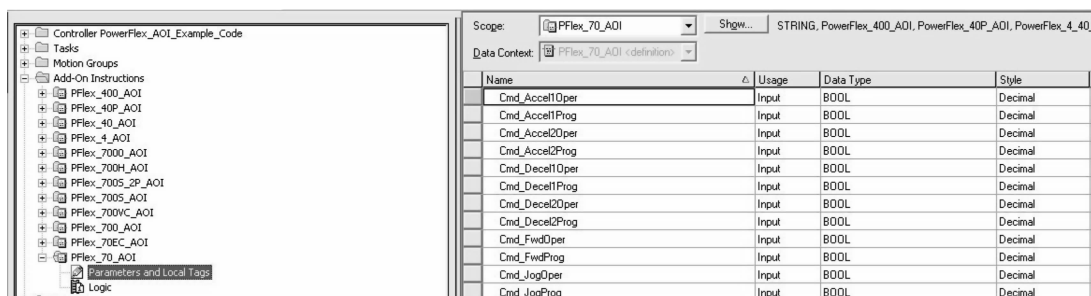


图 24-23　参数和本地标签

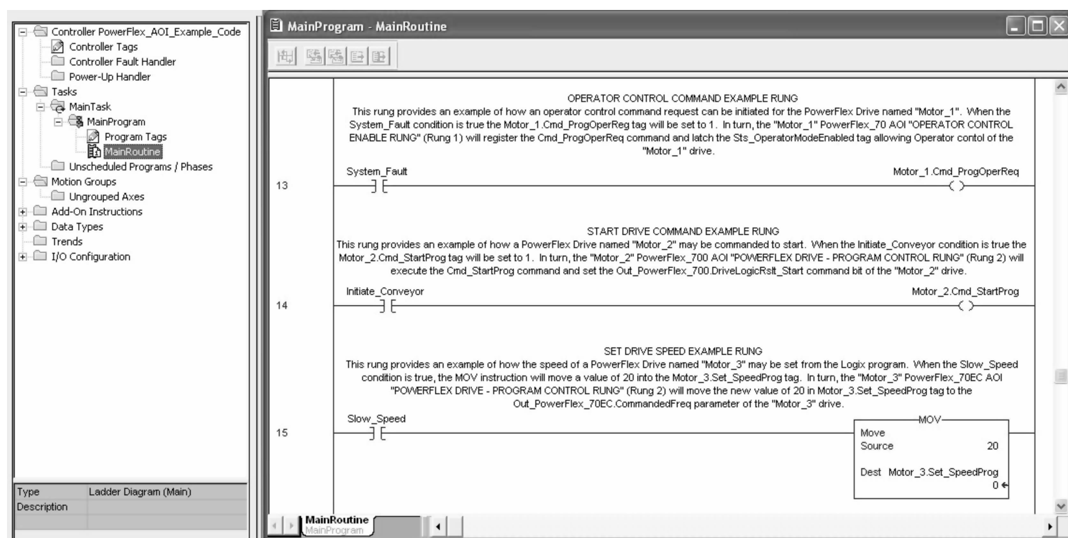


图 24-24　控制程序

2）AOI 程序和操作员控制切换的考量因素：

每个 AOI 的前三个梯级都会处理从 Logix 程序控制（程序控制）到面板控制（操作员控制）的切换。下面显示了两个操作员控制梯级。了解这些梯级的运行方式非常重要，只有这样才可以实现从所需程序控制到操作员控制的切换，以及从操作员控制到程序控制的切换，如图 24-25 所示。

以上面显示的梯级为例，启用预定的 PowerFlex 变频器的操作员控制。可以通过 Logix 程序（ProgOperReq）或 RSViewME/SE PowerFlex 面板（OperOperReq）启动命令请求（Cmd）。

特定的变频器示例：

要为与“Motor_1”AOI 实例关联的变频器启用操作员控制，可在特定的输入条件能够启动标签名称为“Motor_1. Cmd_ProgOperReq”的输出指令的情况下，将梯级添加到 LogixMainRoutine。同样，如果 RSView ME/SE 面板上的按钮#1 参数的值标签名称为“#1. Cmd_OperOperReq”，则可在与“Motor_1”标签关联的面板显示界面中添加。

“Motor_1. Cmd_ProgToOperPermissive”必须也包括在 Logix 程序中，以对允许还是操作“程序控制到操作员控制”切换请求进行控制。标签值 1 表示允许相关的控制切换请求。标签值 0 表示禁止相关的控制切换请求。

注意：

如果 Xxxx. Cmd_ProgToOperPermissive 被设置为 1，则相关的 AOI 会被设定为允许从程序控制切换到操作员控制，而不管 PowerFlex 变频器的当前方向、速度或活动状态如何。此外，AOI 还会被设定为将当前程序速度值“Xxxx. Set_SpeedProg”切换到关联的操作员速度标签“Xxxx. Set_SpeedOper”，从而在进行从程序控制到操作员控制的切换时，创建“无扰动”切换。因此，如果 PowerFlex 变频器在从程序控制转换到操作员控制时正在运行，则 PowerFlex 变频器将继续以最新指定的程序速度运行。如果需要进行从不同程序到操作员控制的转换，则必须监视 PowerFlex 变频器和操作员命令请求的状态，并且必须在允许 Xxxx. Cmd_ProgToOperPermissive 之前发出相应的程序命令。因此，对 PowerFlex 变频器的状态进行监视，并在设置 Xxxx. Cmd_ProgToOperPermissive 标签之前发出相应的程序命令（针对特定应用）这一点非常重要，可以确保程序控制安全地切换到操作员控制。

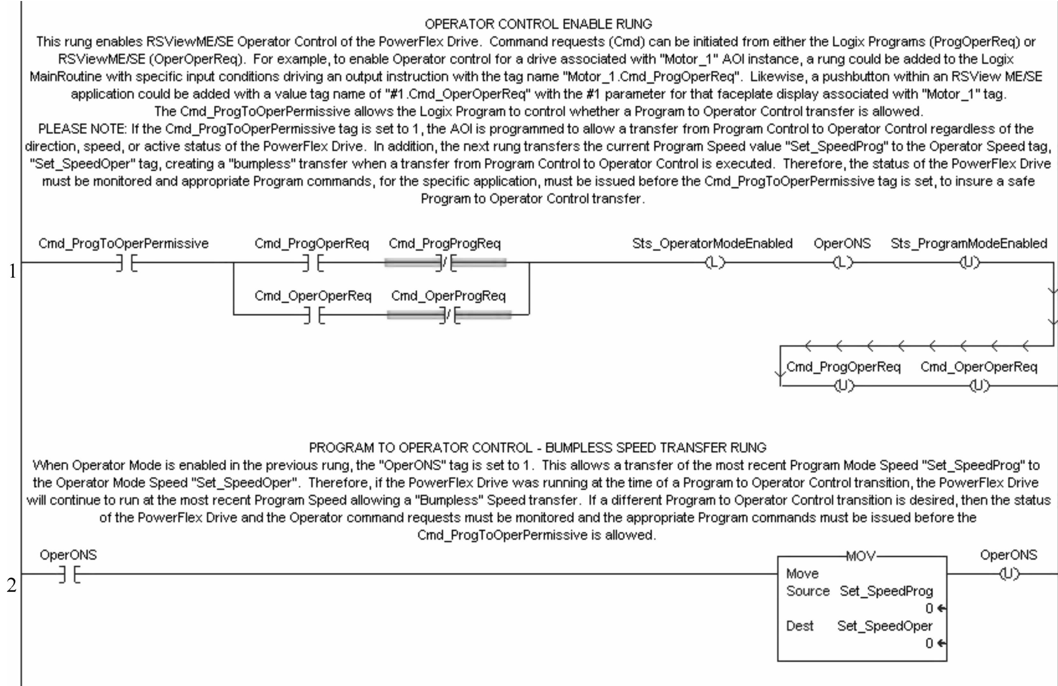


图 24-25 控制程序

注意：

如果 Xxxx. Cmd_OperToProgramPermissive 标签被设置为 1，则关联的 AOI 会被设定为允许从操作员控制切换到程序控制，而不管 PowerFlex 变频器的当前方向、速度或活动状态如何。此外，AOI 中没有设定在切换为程序模式后设置“Set_SpeedProg”的逻辑，因此如果没有在 AOI 之外设定逻辑，则 PowerFlex 变频器会采用上次设定的“Set_SpeedProg”速度值，除非逻辑对 AOI 编程以设置不同的值。因此，对 PowerFlex 变频器的状态进行监视，并在设置 Xxxx. Cmd_OperToProgramPermissive 标签之前发出相应的程序命令（针对特定应用）这一点非常重要，可以确保操作员控制安全地切换到程序控制。

注意：

如果 PowerFlex 变频器是在操作员模式中启动，且 RSView ME PowerFlex 面板在 PowerFlex 变频器正在运行时关闭，则 PowerFlex 变频器将继续以当前的 Set_SpeedOper 标签值运行。因此，了

解该操作如何影响特定应用以及如何执行相应的操作和安全防护措施，这一点非常重要。

注意：

AOI 中没有对已发送到特定变频器的最小或最大 PowerFlex 给定速度值进行控制的逻辑。因此，了解已连接到特定变频器的电动机的限制、设置相应 PowerFlex 变频器参数以及添加附加程序逻辑（如有必要）是非常重要的。

二、RSViewME 组态

1. 打开要将 PowerFlex 面板添加到其中的现有 RSView ME 应用程序。

2. 将所需“PowerFlex_xxx_Faceplate. gfx”添加到显示界面。

1）右键单击“显示界面”（Displays），然后选择“将组件添加到应用中”（Add Component Into Application）。

2）浏览至 ME_PowerFlex_Faceplate_Files 文件夹，并选择要使用的面板 GFX 文件，如图 24-26 所示。



图 24-26 添加组件

3. 为 RSViewME 应用程序中的 ME_PowerFlex_Parameter. par 添加面板参数文件。

1）右键单击“参数”（Parameters），然后选择“将组件添加到应用中”（Add component to application）。

2）浏览至 ME_PowerFlex_Faceplate_Files 文件夹，然后选择 ME_PowerFlex_Parameter. par 文件，并单击“打开”（Open）按钮，如图 24-27 所示（通常，您可使用与要监视和控制的预定变频器相关联的名称重新命名该参数）。

3）现在，打开刚刚添加的参数，然后编辑参数“tagname”，使其与应用匹配。参数“tagname”需要包括系统控制器的 RSLinx Enterprise“设备快捷方式”名称，以及要监视和控制的预定 PowerFlex 变频器的“模块名称”。

a）双击面板参数文件，然后编辑“设备快捷方式”名称，以与先前在 RSLinx Enterprise 通信设置中组态的 Logix 控制器（如 CLX-L63）匹配，如图 24-28 所示。

b）输入您需要监视和控制的、先前已在步骤 3b 中，于 Logix 组态部分内组态的预定 PowerFlex 变频器（即 Motor_1）的所需“模块名称”。关闭并保存更改，如图 24-29 所示。

4）要为其他所要监视和控制的 PowerFlex 变频器添加参数文件，仅需重复上述步骤 a 和步骤 b，以创建不同的参数名称并为参数“tagname”分配关联的“设备快捷方式”和“模块名称”

即可。

4. 创建显示界面导航，以在运行时打开 PowerFlex 面板（要使用预先创建的 PowerFlex“转到”（Goto）按钮，请跳至第 19 页，参阅“使用预先创建的转到按钮的组态步骤”）。

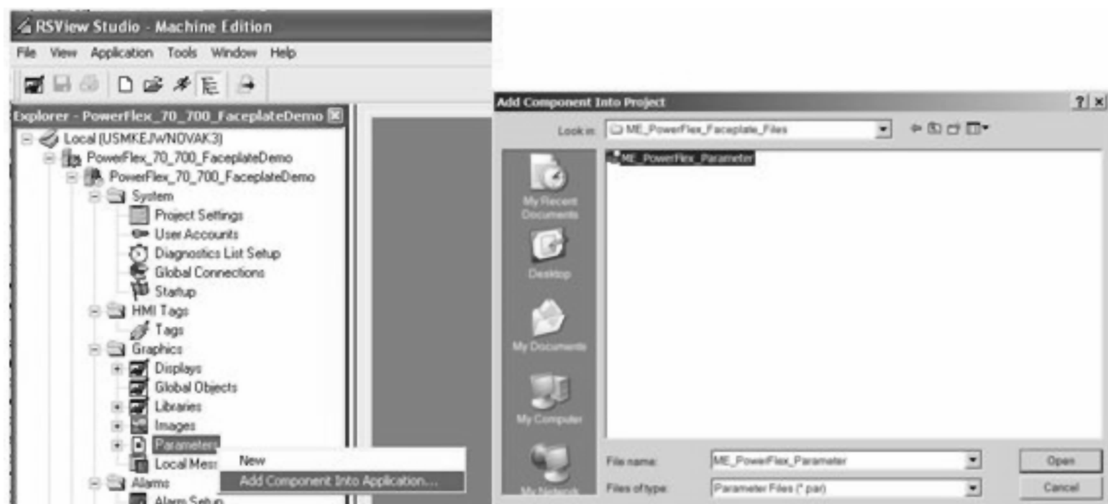


图 24-27 添加面板参数文件

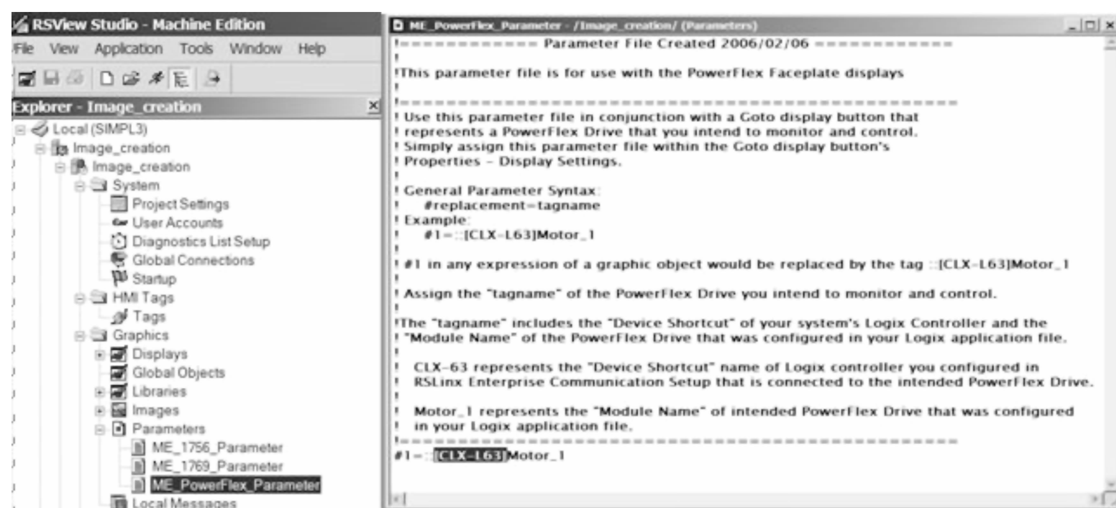
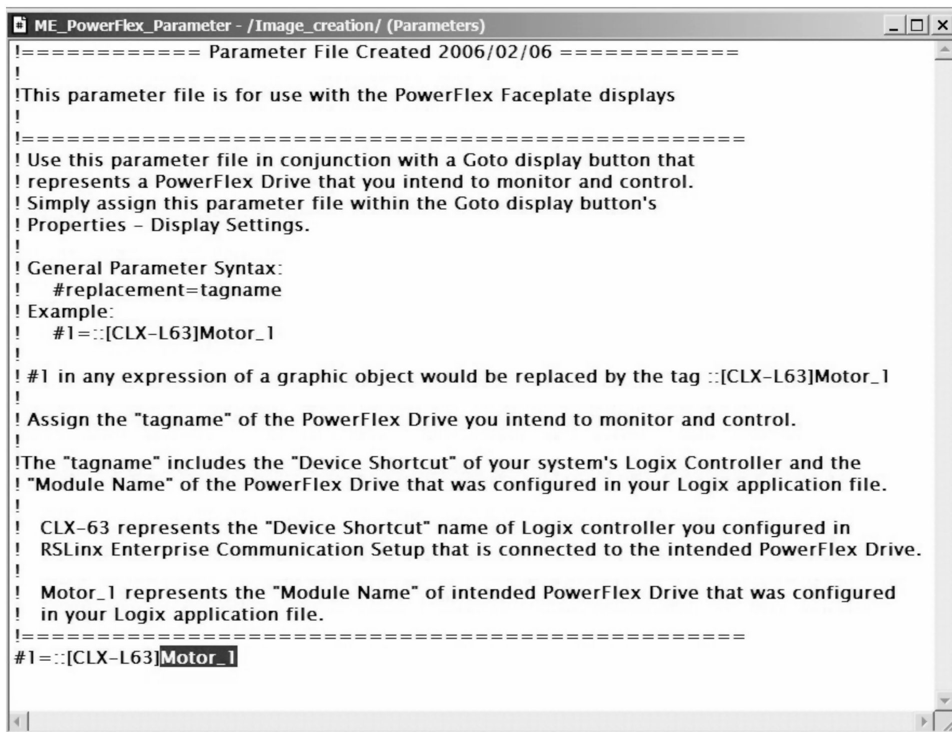


图 24-28 编辑“设备快捷方式”名称

1）在所需图形显示界面中，于代表您打算监视和控制的电动机/变频器的图形对象之上，创建新的透明“转到”（Goto）显示界面导航按钮。双击“转到”（Goto）按钮，以对其“显示设置”（Display settings）进行组态。选择“显示”（Display）字段旁的浏览按钮，并分配 PowerFlex 面板显示界面，如图 24-30 所示。

2）选择“参数文件”（Parameter file）字段旁的浏览按钮，并为该预定变频器分配之前创建的 PowerFlex 参数文件，如图 24-31 所示。

3）要支持其他 PowerFlex 变频器，仅需创建新的“转到”（Goto）显示界面导航按钮、分配 PowerFlex 面板显示界面，以及分配与其他 PowerFlex 变频器关联的唯一参数文件。



ME_PowerFlex_Parameter - /Image_creation/ (Parameters)

```
!============ Parameter File Created 2006/02/06 ============
!
!This parameter file is for use with the PowerFlex Faceplate displays
!
!=================================================
! Use this parameter file in conjunction with a Goto display button that
! represents a PowerFlex Drive that you intend to monitor and control.
! Simply assign this parameter file within the Goto display button's
! Properties - Display Settings.
!
! General Parameter Syntax:
!    #replacement=tagname
! Example:
!    #1=::[CLX-L63]Motor_1
!
! #1 in any expression of a graphic object would be replaced by the tag ::[CLX-L63]Motor_1
!
! Assign the "tagname" of the PowerFlex Drive you intend to monitor and control.
!
!The "tagname" includes the "Device Shortcut" of your system's Logix Controller and the
! "Module Name" of the PowerFlex Drive that was configured in your Logix application file.
!
!  CLX-63 represents the "Device Shortcut" name of Logix controller you configured in
!  RSLinx Enterprise Communication Setup that is connected to the intended PowerFlex Drive.
!
!  Motor_1 represents the "Module Name" of intended PowerFlex Drive that was configured
!  in your Logix application file.
!=================================================
#1=::[CLX-L63]Motor_1
```

图 24-29　输入“模块名称”

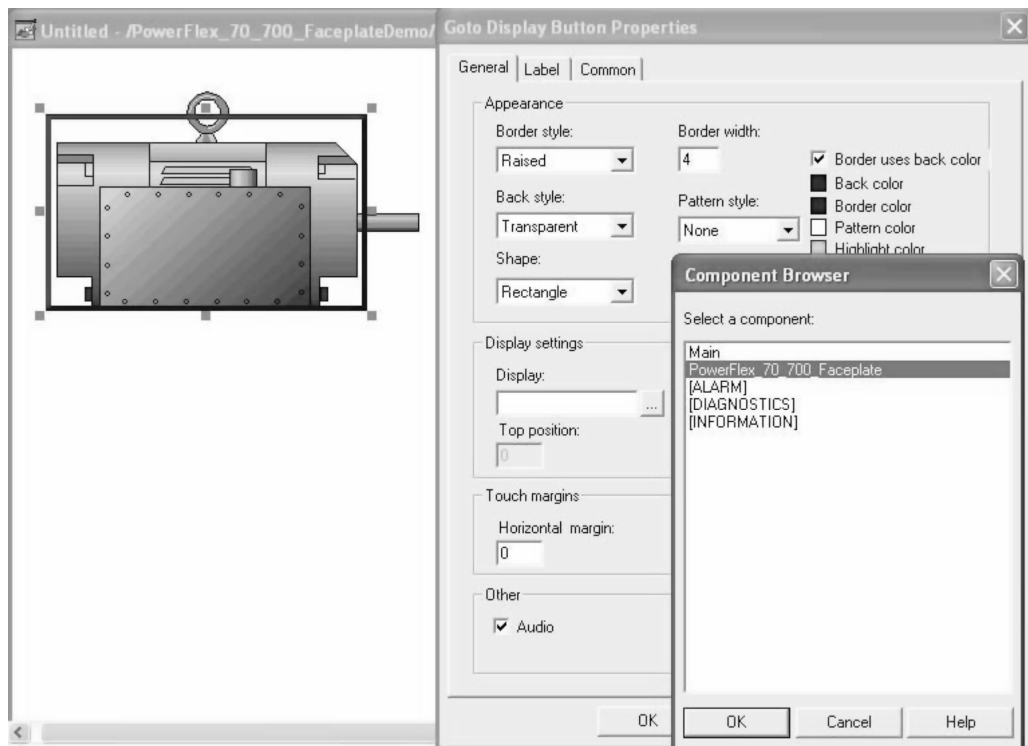


图 24-30　配置“转到”按钮

三、使用预先创建的转到按钮的组态步骤

如果希望硬件系统视图如图 24-32 所示，将 PowerFlex 变频器对象组态为可以启动单独 PowerFlex 面板的“转到”按钮，请按照以下说明进行操作。

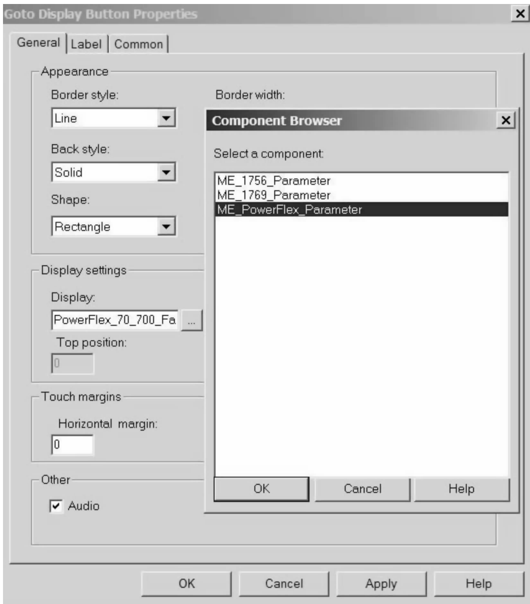


图 24-31　变频器分配 PowerFlex 参数文件



图 24-32　硬件系统视图

1. 通过右键单击系统树中的“图像”（Image）文件夹，并选择“将组件添加到应用中”（Add Component into Application），将 PowerFlex 位图图像添加到 RSView 应用中，如图 24-33 所示。

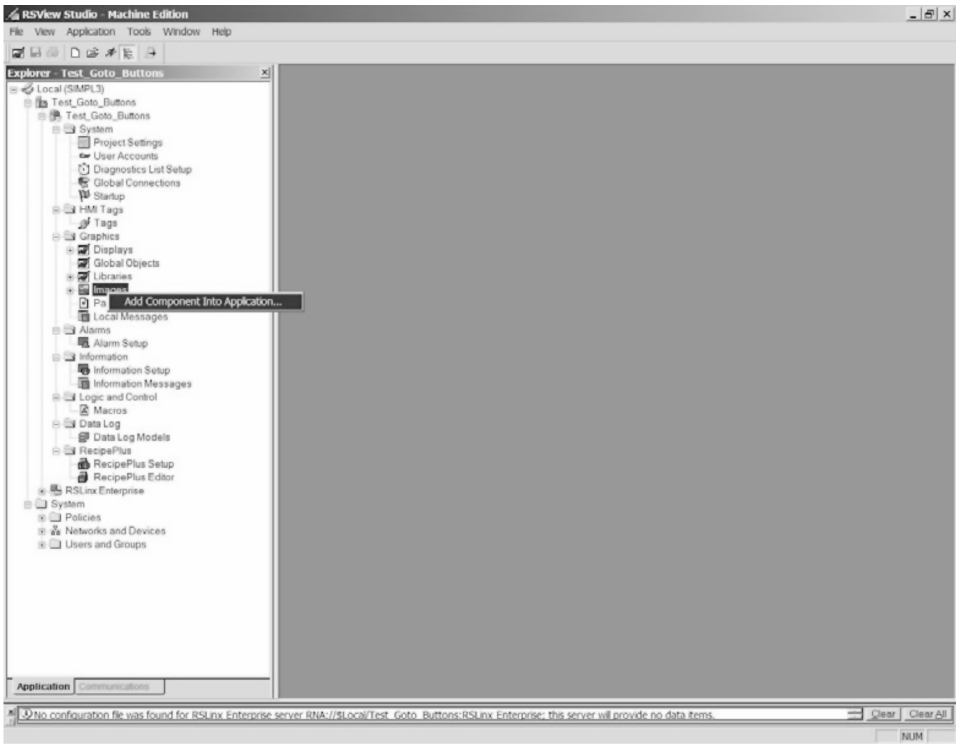


图 24-33　添加 PowerFlex 位图图像

2. 解压缩 ME_PowerFlex_Faceplate_Files. zip 文件时浏览至步骤 1 中所创建的 “ME_PowerFlex_GotoButtons” 文件夹。选择所有位图图像并单击 “打开”（Open），如图 24-34 所示。

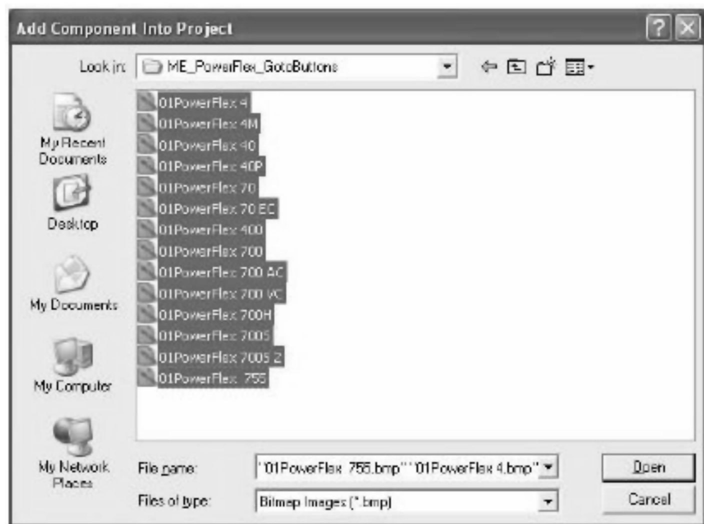


图 24-34 打开所有位图图像

3. 接下来，通过右键单击系统树中的 “显示界面”（Display）文件夹，并选择 “将组件添加到应用中”（Add Component into Application），将 “PowerFlex 转到显示界面” 文件添加到 RS-View 应用中，如图 24-35 所示。

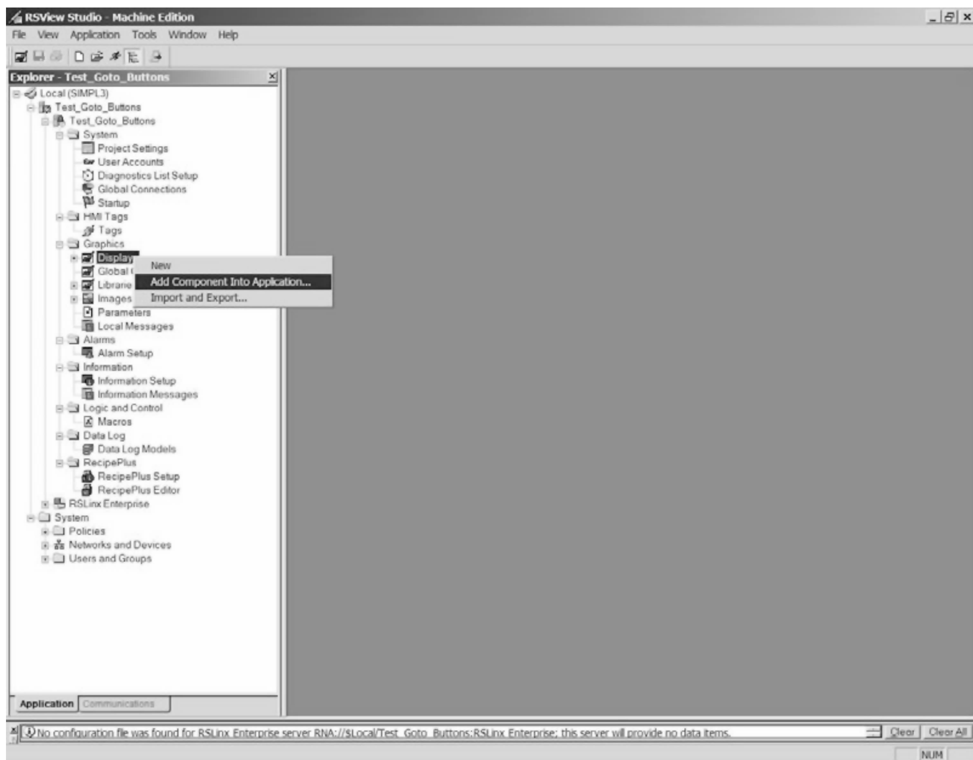


图 24-35 添加 “PowerFlex 转到显示界面” 文件

4. 浏览至“ME_PowerFlex_GotoButtons”文件夹，并选择“Goto_PowerFlex.gfx”文件。单击“打开”（Open）。检验“Goto_PowerFlex”显示界面是否出现在系统树中的“显示界面”（Displays）下方。

5. 创建新显示界面，或打开要将“转到”（Goto）按钮复制到其中的您自己的系统显示界面。

6. 打开“Goto_PowerFlex”显示界面，并将所需图形对象复制（或者拖放）到系统显示界面上，如图24-36所示。

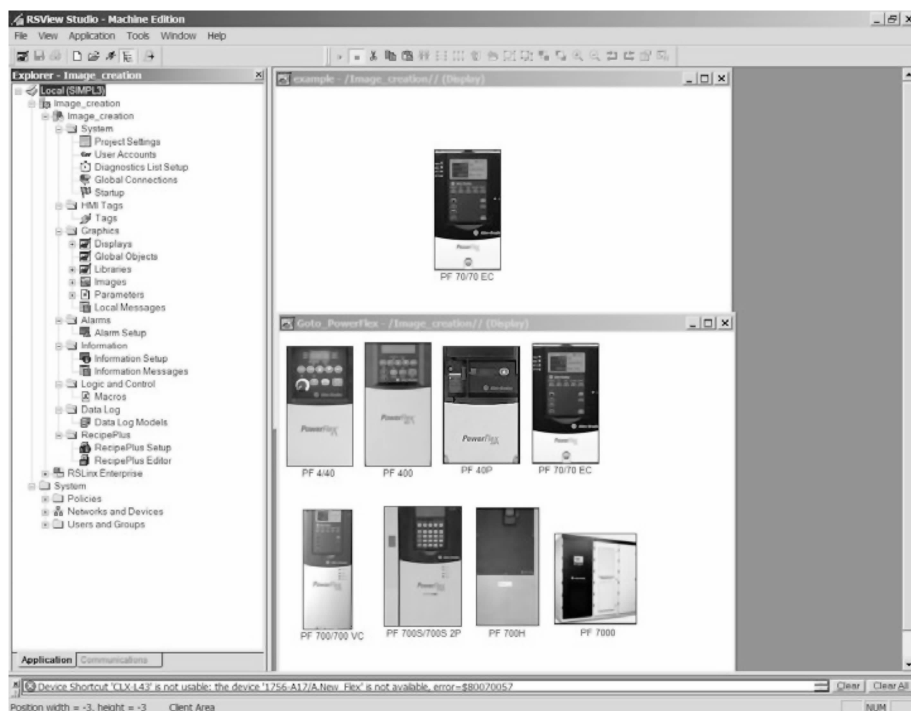


图24-36 复制所需图形对象

7. 重复步骤8，直至将所有需要使用的对象都复制到显示界面上。

8. 通过双击对象，打开每一个模块“转到”（Goto）显示界面按钮（请务必单击对象顶部附近的位置，以打开“转到”（Goto）按钮属性，而非“文本”对象属性）。

9. 在“常规”（General）选项卡上，分配PowerFlex面板和参数文件，使其与该按钮关联。

1）选择“显示”（Display）旁的浏览按钮，并分配PowerFlex面板显示界面，如图24-37所示。

2）选择“参数文件”（Parameter file）旁的浏览按钮并分配PowerFlex参数文件。请务必单击“转到显示界面按钮属性”（Goto Display Button Properties）对话框的“确定”（OK）按钮，以保存组态，如图24-38所示。

10. 可通过选择对象并用鼠标进行缩小或放大操作，来调整对象的大小（要调整多个对象的大小，可通过使用鼠标在所有对象周围绘制方框以选择所有对象，然后从菜单中选择“排列 > 分组”（Arrange > Group）的方式，对所有对象进行分组。使用鼠标即可调整对象的大小）。

11. 所有需要的PowerFlex对象均已复制到应用中之后，可以将“Goto_PowerFlex”显示界面从应用中删除。

1）在左侧的系统树中，右键单击显示界面下的“转到”（Goto）显示界面，然后选择删除。

2）单击“是”（Yes），会弹出“从应用中移除文件并将其从磁盘中删除”（Remove file from the application and delete it from disk）。

12. 现在可以保存并创建运行时文件，以测试显示界面。

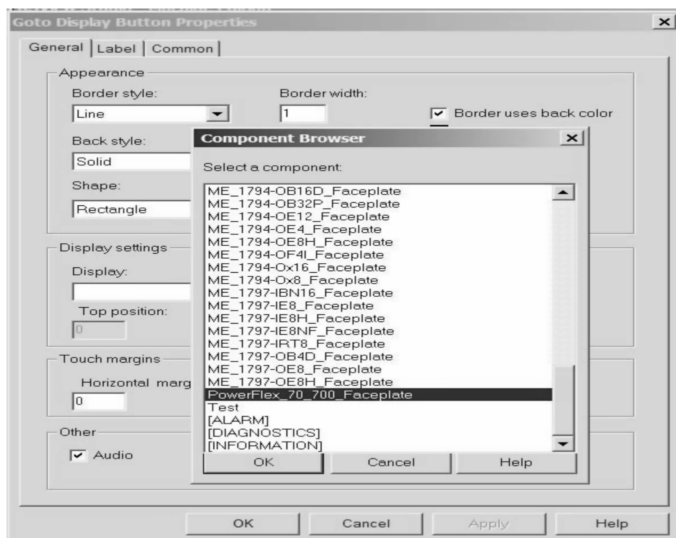


图 24-37　分配 PowerFlex 面板显示界面

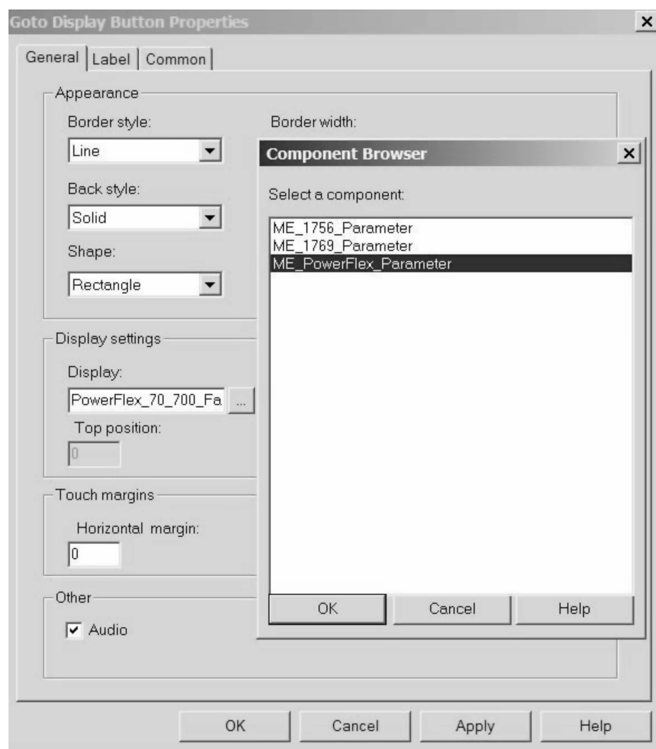


图 24-38　分配 PowerFlex 参数文件

第六部分　变频器的首要集成

实验 25：PowerFlex 70 变频器源代码级集成

一、实验目的

创建包含 PowerFlex 集成变频器配置文件的 RSLogix 5000 项目，并介绍如何下载到变频器和控制器以及如何确认网络 I/O 可正常运行。

二、创建 RSLogix 5000 项目

1. 启动程序。双击桌面上或“开始”（Start）菜单中的 RSLogix 5000 图标，选择“所有程序 > 罗克韦尔软件 > RSLogix 5000 企业版系列 > RSLogix 5000”（All Programs > Rockwell Software > RSLogix 5000 Enterprise Series > RSLogix 5000）。

2. 创建新项目。选择“文件 > 新建”（File > New）。输入 ControlLogix 演示设备插槽 0 中控制器的类型（例如 1756-L63），并输入如图 25-1 所述的所有其他数据。单击“确定”（OK）。

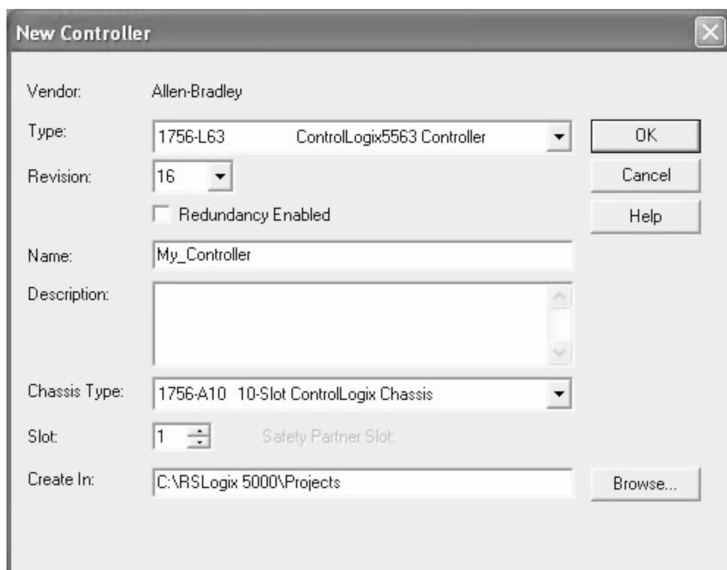


图 25-1　新控制器

可能会出现一个窗口，提示计算机上已存在此文件名。如果您使用的名称与本示例中的名称相同，请单击“是”（Yes）覆盖此文件。

集成变频器配置文件几乎可用于所有驱动器，并随 RSLogix 5000 软件的版本 16（或更高版本）一起提供。

3. 向系统添加新模块。右键单击“I/O 组态”（I/O Configuration）文件夹下的 1756 背板并选择“新建模块”（New Module），如图 25-2 所示。

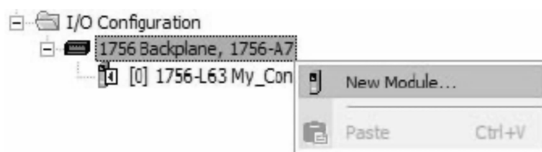


图 25-2　新建模块

该操作会打开“选择模块”（Select Module）窗口，该窗口会列出 1756 背板的可用模块，如图 25-3 所示。

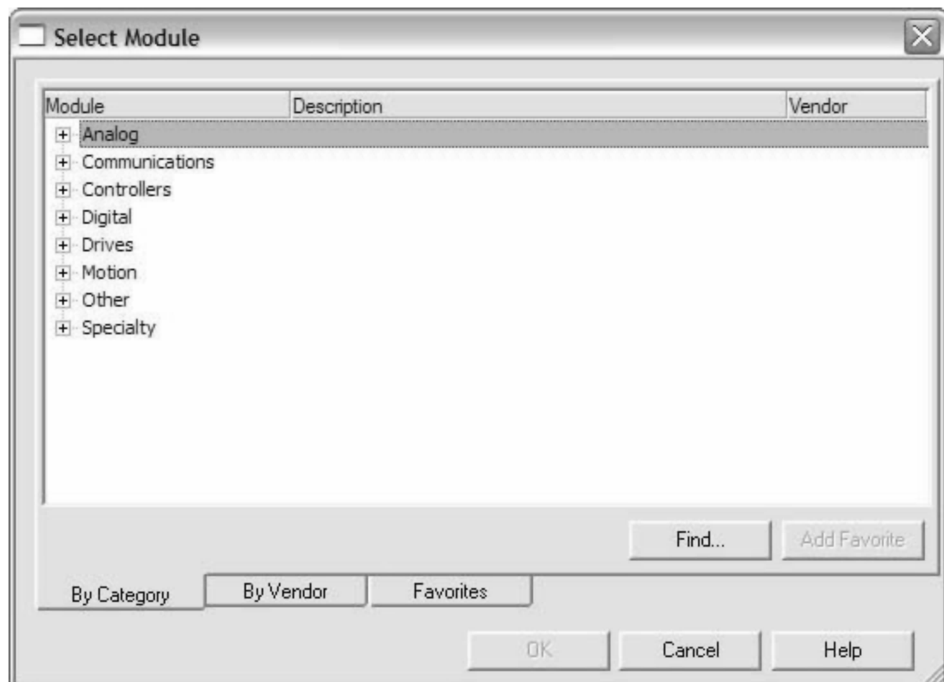


图 25-3　模块选择

4. 向系统添加“以太网网桥”，展开“通信”（Communications）部分，向下滚动并选择 1756-ENBT/A，然后单击“确定”（OK），如图 25-4 所示。

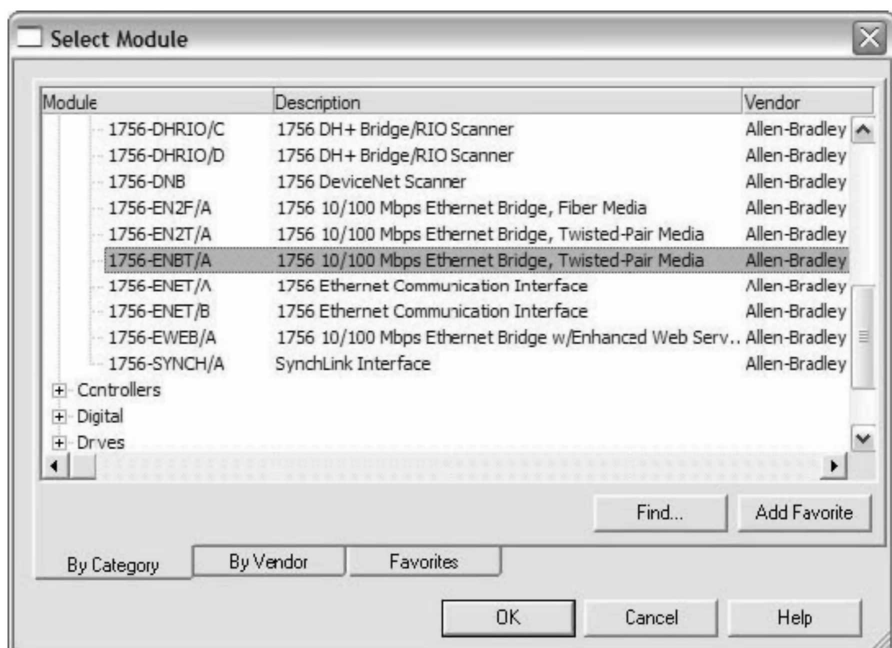


图 25-4　添加通信模块

5. 选择 ENBT 的版本。选择 3 作为主版本，然后单击“确定”（OK），如图 25-5 所示。

6. 输入 ENBT 的信息。输入 ENBT 数据，如图 25-6 所示。

如果选中了“打开模块属性”（Open Module Properties）框，则应取消选择该框，然后单击“确定”（OK）关闭“新建模块”（New Module）窗口。

7. 验证 ENBT 是否位于 I/O 组态文件夹中。现在，ENBT 应出现在“I/O 组态”（I/O Configuration）文件夹中的 1756 背板下方，如图 25-7 所示。

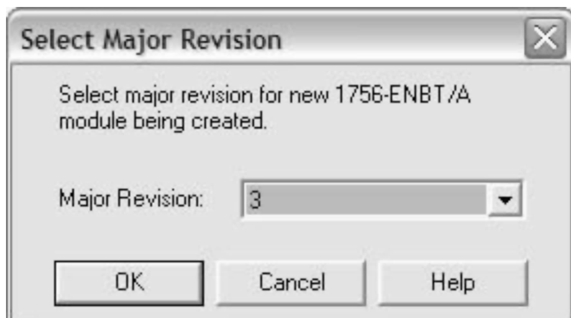


图 25-5　选择主版本

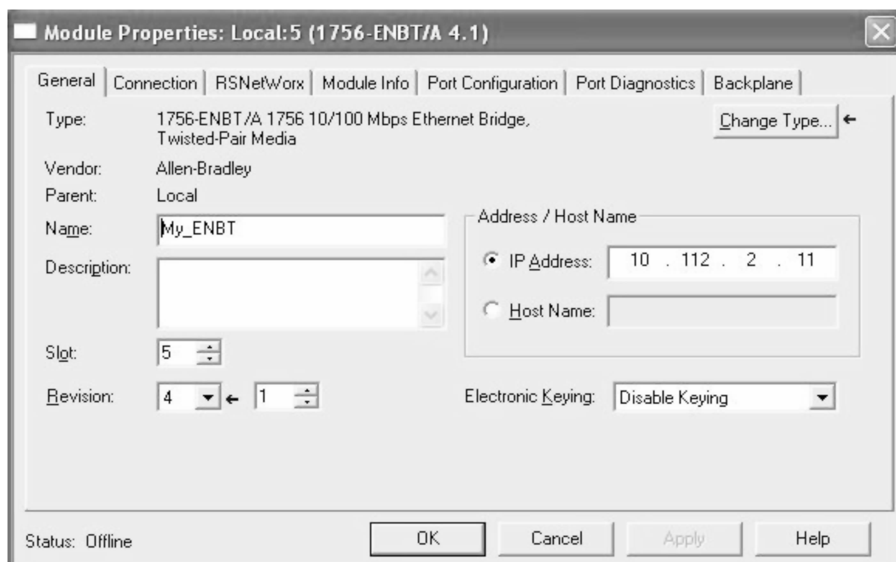


图 25-6　配置通信模块

8. 向系统添加新模块。右键单击“I/O 组态”（I/O Configuration）文件夹下的 1756-ENBT 并选择“新建模块”（New Module），如图 25-8 所示。

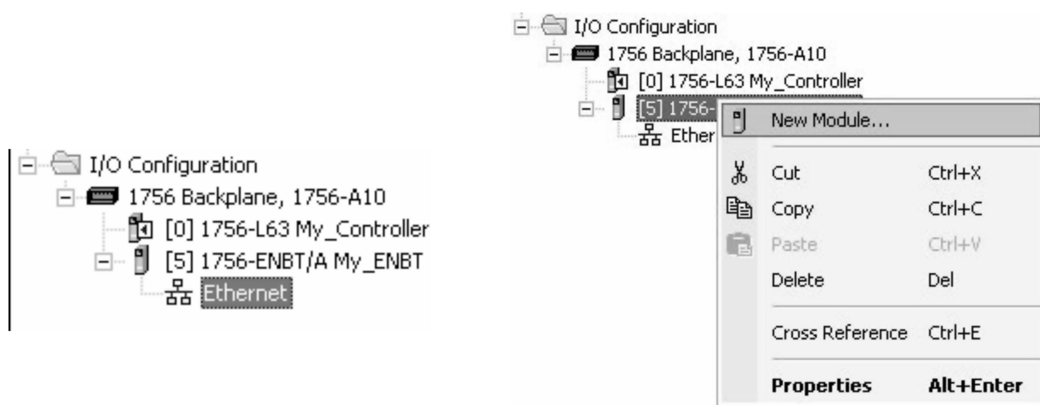


图 25-7　I/O 组态列表

图 25-8　添加新模块

该操作会打开“选择模块”（Select Module）窗口，该窗口会列出 ENBT 的可用模块，如图 25-9 所示。

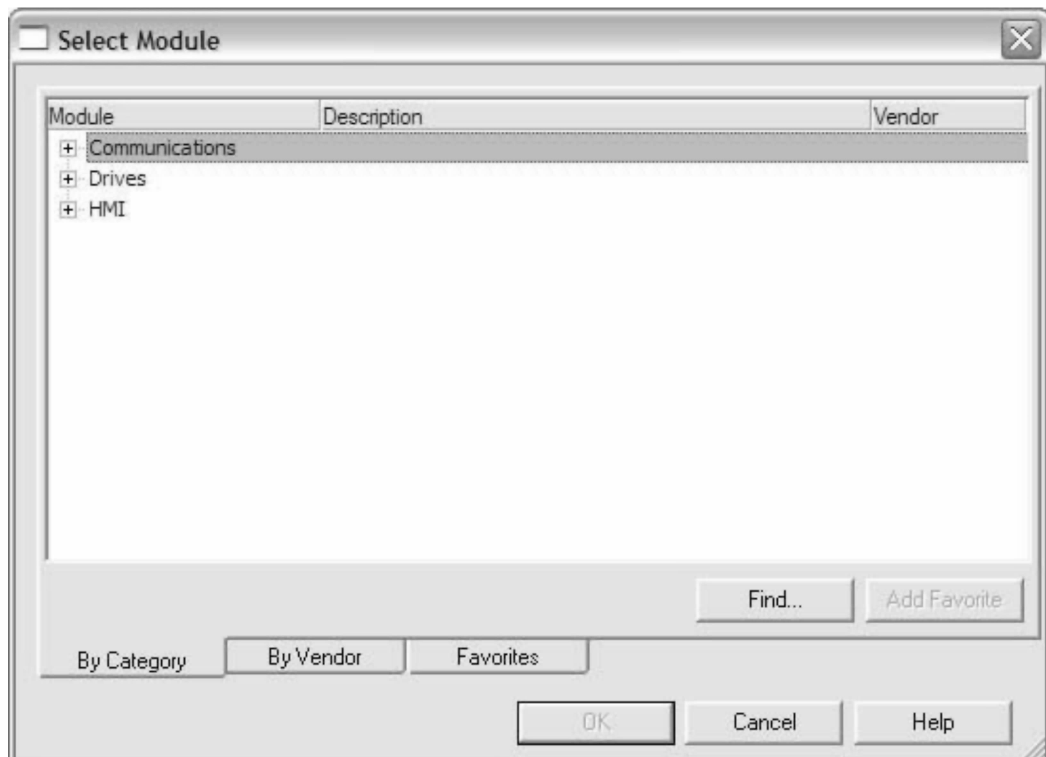


图 25-9　选择模块

9. 向系统添加 PowerFlex 70 变频器。展开“变频器”（Drives）部分，向下滚动并查看目前版本 V16 中可用的新集成变频器配置文件。之前必须使用“通用”配置文件的多个变频器（如 PowerFlex 4 类变频器）现在有了自己特有的配置文件。选择 PowerFlex 70-E，然后单击“确定”（OK），如图 25-10 所示。

10. 输入 PowerFlex 70 节点信息。输入 PowerFlex 70 的名称和 IP 地址数据，如图 25-11 所示。

现在，请单击“模块定义”（Module Definition）部分中的“更改”（Change），编辑 I/O 连接信息。将出现“模块定义”（Module Definition）窗口，如图 25-12 所示。

将变频器的主版本设为 2，次版本设为 2，“电子匹配”（Electronic Keying）设为“禁用电子匹配功能”（Disable Keying）。将变频器额定值设置为 240V 4.2A，如图 25-12 所示。不要单击“确定”（OK）。

最多可为连接定义 10 个字的输入数据和 10 个字的输出数据。在默认情况下，将会传送变频器状态和控制信息。

状态信息“DriveStatus”字包含逻辑状态位信息、如就绪、报警、故障和达到基准值。“OutputFreq”字包含反馈信息，该信息可显示变频器的实际工作频率。

控制信息“DriveLogicRslt”字包含逻辑命令位信息，如停止、开始、正向、反向和重置。“CommandedFreq”字包含用于使变频器运行的给定基准值。

数据链路 A、B、C、D 数据链路是指向变频器参数的指针。我们的变频器 I/O 组件并不是所

见即所得的固定 I/O 组件，而是动态的组件，并具有使程序员能够挑选并选择所需变频器参数的功能，从而可以作为网络 I/O 进行通信。

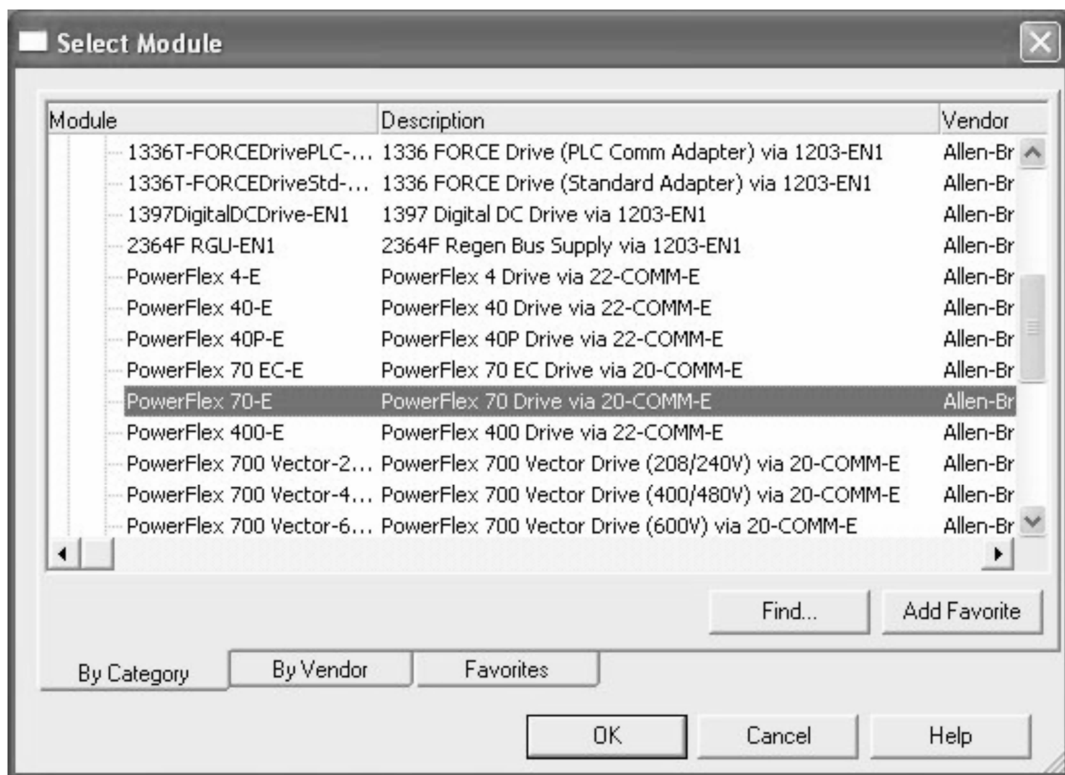


图 25-10　选择变频器型号

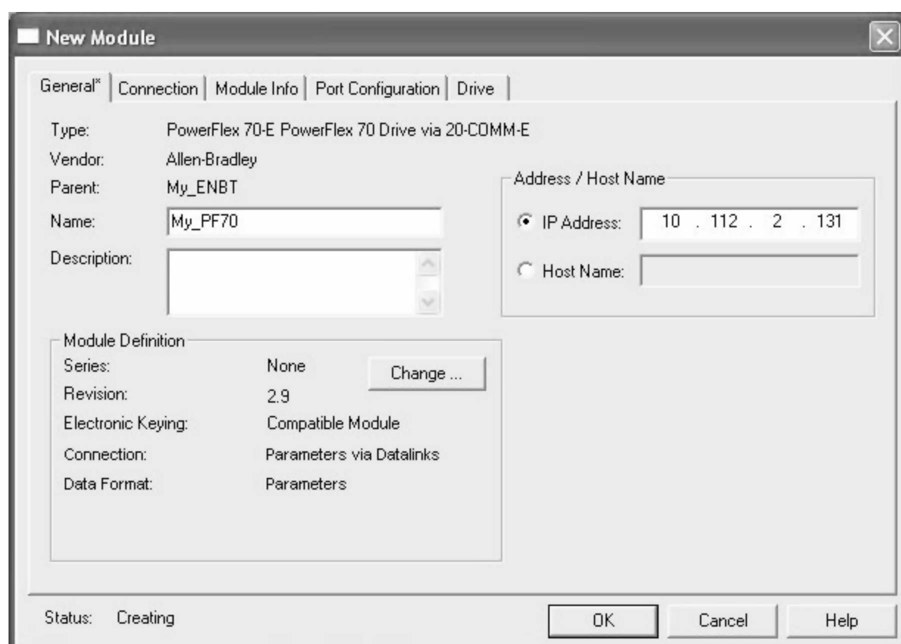


图 25-11　输入变频器名称和 IP 地址

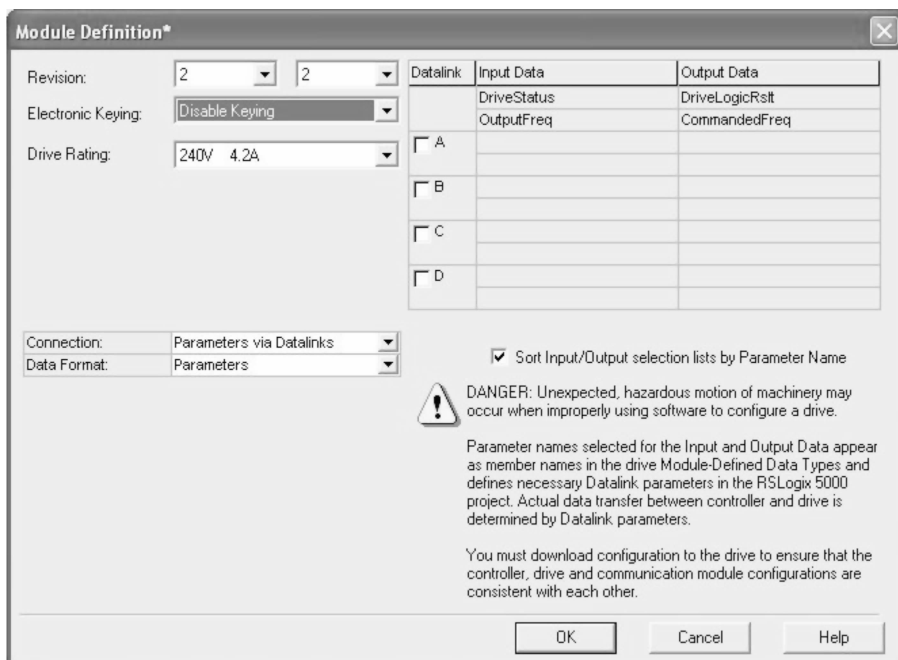


图 25-12 “模块定义”图

11. 启用数据链路 A 输入数据。选中数据链路 A 框，以启用数据链路 A 信息，从而能够以网络 I/O 的形式进行通信。但不要单击“确定”（OK），如图 25-13 所示。

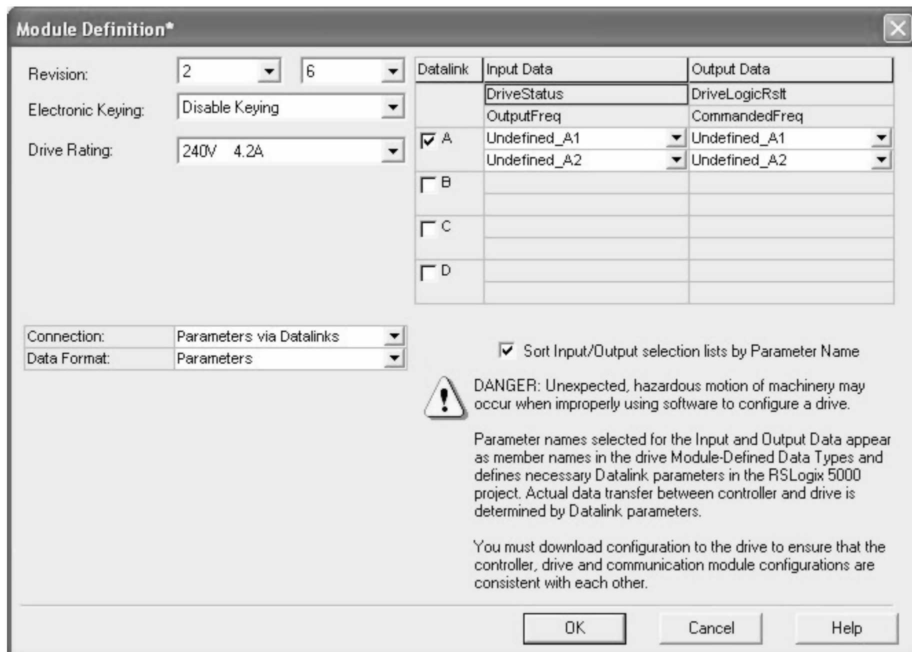


图 25-13 定义数据链路

启用数据链路后，最多允许通过网络 I/O 读取两个 16 位参数并写入两个 16 位参数。选中数据链路 A 后，还会自动为 20-COMM-E EtherNet/IP 适配器设置以下参数：

1）参数 23［DPII/O 组态］－置位数据链路 A 位，这样，20-COMM-E 将与变频器传送数据链路 A 信息。

2）参数 35［M-S 输入］－置位数据链路 A 位，这样，20-COMM-E 会由控制器输入数据链路 A 信息。

3）参数 36［M-S 输出］－置位数据链路 A 位，这样，20-COMM-E 会将数据链路 A 信息输出到控制器。

这一点突显了 PowerFlex 集成变频器配置文件的优势之一：使用方便。本示例中，通过一个复选框简化了操作，用户不必编辑三个不同参数并置位每个参数中的特定位。

12. 组态数据链路 A 的输入数据。

单击“输入数据”（Input Data）列中的 Undefined_ A1下拉箭头，查看 PowerFlex 参数。选择 AccelTime1。

单击“输入数据”（Input Data）列中的 Undefined_ A2下拉箭头，查看 PowerFlex 参数。可以滚动到 DecelTime1 参数，然后选择该参数，还可以在该框中输入“de”，以显示以这些字符开头的参数。使用其中一种方法为 Datalink A2 输入 DecelTime1。

不要单击“确定”（OK），（见图 25-14）。

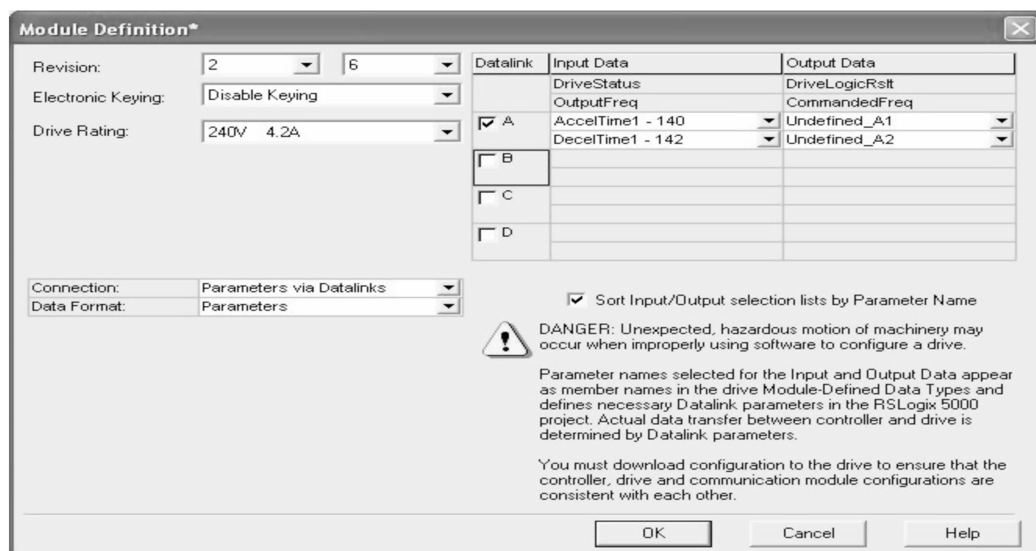


图 25-14 组态数据链路 A 输入数据

输入数据链路 A 的输入数据之后，会自动为 PowerFlex 70 设置以下参数：

1）参数 310［数据输出 A1］－设置为 140，这样，当数据链路 A1 信息发送到 20-COMM-E 时，PowerFlex70 将发送参数 140［加速时间 1］。

2）参数 311［数据输出 A2］－设置为 142，这样，当数据链路 A2 信息发送到 20-COMM-E 时，PowerFlex70 将发送参数 142［减速时间 1］。

13. 组态向数据链路 A 输出的数据。

单击“输出数据”（Output Data）列中的 Undefined_ A1下拉箭头，查看 PowerFlex 参数。选择 AccelTime1。

单击“输出数据”（Output Data）列中的 Undefined_ A2下拉箭头，查看 PowerFlex 参数。可以滚动到 DecelTime1 参数，然后选择该参数，还可以在该框中输入“de”，以显示以这些字符开

头的参数。使用其中一种方法为 Datalink A2 输入 DecelTime1，如图 25-15 所示。

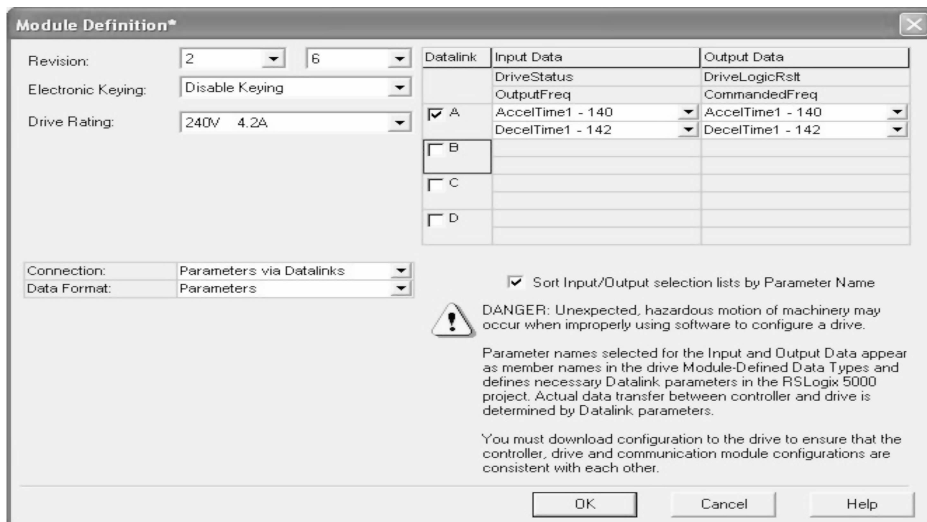


图 25-15　组态数据链路 A 输出数据

不要单击“确定”（OK）。

输入数据链路 A 的输出数据之后，会自动为 PowerFlex 70 设置以下参数：

1）参数 300［数据输入 A1］－设置为 140，这样，PowerFlex 70 将接收 20-COMM-E 发送的数据链路 A1 信息，并将其写入参数 140［加速时间 1］。

2）参数 301［数据输入 A2］－设置为 142，这样，PowerFlex 70 将接收 20-COMM-E 发送的数据链路 A2 信息，并将其写入参数 142［减速时间 1］。

14. 组态数据链路 B 的输入数据。选中数据链路 B 框，以便能够以网络 I/O 的形式传送数据链路 B 信息。

单击“输入数据”（Input Data）列中的 Undefined_ B1下拉箭头，查看 PowerFlex 参数。可以滚动到 FaultCode1 参数，然后选择该参数，还可以在该框中输入“f”，以显示以这些字符开头的参数。使用其中一种方法为 Datalink B1 输入 FaultCode1。

单击“输入数据”（Input Data）列中的 Undefined_ B2下拉箭头，查看 PowerFlex 参数。可以滚动到 DCBusVoltage 参数，然后选择该参数，还可以在该框中输入“dc”，以显示以这些字符开头的参数。使用其中一种方法为 Datalink B2 输入 DCBusVoltage，如图 25-16 所示。

不要单击“确定”（OK）。

选中数据链路 B 后，会自动为 20-COMM-E EtherNet/IP 适配器设置以下参数：

1）参数 23［DPII/O 组态］－置位数据链路 B 位，这样 20-COMM-E 将与变频器传送数据链路 B 信息。

2）参数 35［M－S 输入］－置位数据链路 B 位，这样 20-COMM-E 会由控制器输入数据链路 B 信息。

3）参数 36［M－S 输出］－置位数据链路 B 位，这样 20-COMM-E 会将数据链路 B 信息输出到控制器。

本示例中，通过一个复选框简化了操作，用户不必编辑三个不同参数并置位每个参数中的特定位。

输入数据链路 B 的输入数据之后，会自动为 PowerFlex 70 设置以下参数：

1）参数 312［数据输出 B1］–设置为 243，这样当数据链路 B1 信息发送到 20-COMM-E 时，PowerFlex70 将发送参数 243［故障代码 1］。

2）参数 313［数据输出 B2］–设置为 12，这样当数据链路 B2 信息发送到 20-COMM-E 时，PowerFlex70 将发送参数 12［直流母线电压］。

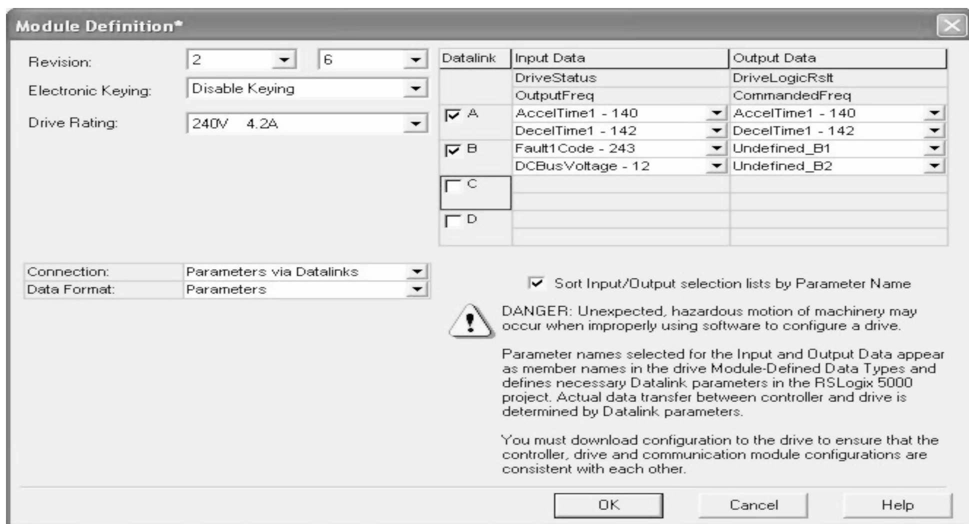


图 25-16　组态数据链路 B 输入数据

15. 组态数据链路 C 的输入数据。选中数据链路 C 框，以便能够以网络 I/O 的形式传送数据链路 C 信息。

单击“输入数据”（Input Data）列中的 Undefined_ C1下拉箭头，查看 PowerFlex 参数。可以滚动到 OutputCurrent 参数，然后选择该参数，还可以在该框中输入“ou”，以显示以这些字符开头的参数。使用其中一种方法为 Datalink C 输入 OutputCurrent。

单击“输入数据”（Input Data）列中的 Undefined_ C2下拉箭头，查看 PowerFlex 参数。可以滚动到 OutputPower 参数，然后选择该参数，还可以在该框中输入“ou”，以显示以这些字符开头的参数。使用其中一种方法为 Datalink C 输入 OutputPower，如图 25-17 所示。

不要单击“确定”（OK）。

本示例中，用户只需要选择所需参数的名称，并不需要了解最终会设置两个不同的参数。

现在，可以单击“确定”（OK），然后单击“警告”（Warning）消息中的“是”（Yes）。I/O 组态完成。

16. 您将再次返回到“常规”（General）窗口。单击“确定”（OK）关闭“模块属性”（Module Properties）窗口。

17. 验证 PowerFlex 70 是否位于 I/O 组态文件夹中。已生成 PowerFlex 70 节点，该节点现在位于 I/O 树中，如图 25-18 所示。

如果您的应用需要使用多个相同的变频器，可以右键单击 PowerFlex 70 复制变频器，然后右键单击 ENBT，根据需要进行多次粘贴。会复制所有节点信息，包括变频器参数设置。所有用户都必须进行的操作是更改 IP 地址并为变频器重新命名。集成变频器配置文件不仅便于使用，还能使更大系统的设计速度更快，如图 25-19 所示。

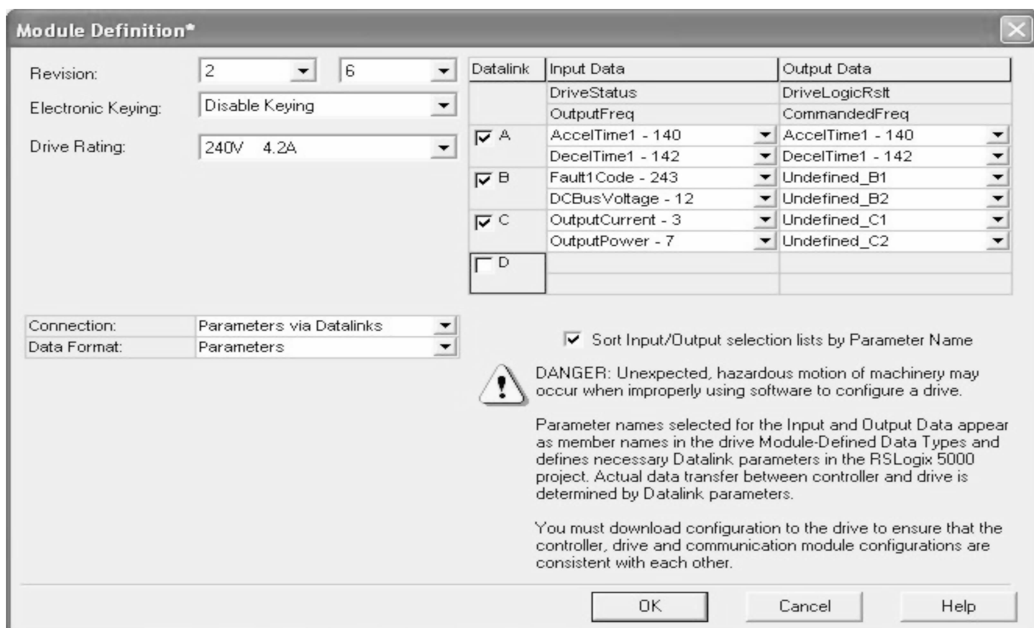


图 25-17 组态数据链路 C 输入数据

您认为变频器组态数据存储在哪里?

每个节点的变频器组态数据实际上存储在您的 RSLogix 5000 项目（ACD 文件）中！将 . ACD 文件下载到控制器时，变频器组态数据就会保存在 ControlLogix 中。这样，如果需要更换节点，就可以为节点的组态设置提供方便的本地资源。只需连接到控制器，然后打开变频器的 AOP 并下载该组态。

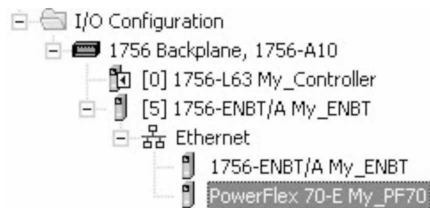


图 25-18 I/O 配置项目列表

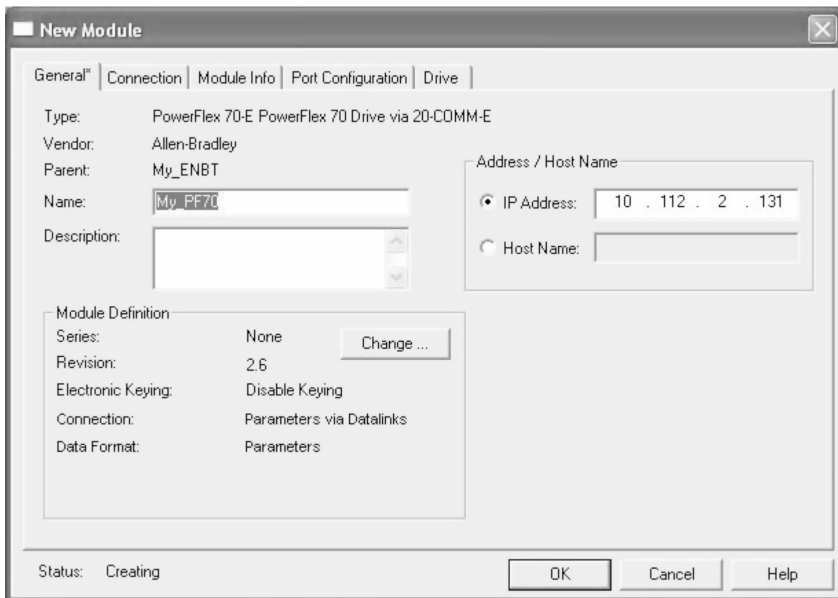


图 25-19 复制变频器参数

18. 查看输入标签。双击“控制器”（Controller）文件夹下的“控制器标签”（Controller Tags），然后展开 My_ PF70E：I 输入。可能需要拖动标题中“名称”（Name）列的右侧边框，以查看完整的输入标签名称，如图 25-20 所示。

Name	Value	Data Type
− My_PF70:I	{...}	AB:PowerFlex70_EEE2565D:I:0
+ My_PF70:I.DriveStatus	2#0000_0000_...	INT
My_PF70:I.DriveStatus_Ready	0	BOOL
My_PF70:I.DriveStatus_Active	0	BOOL
My_PF70:I.DriveStatus_CommandDir	0	BOOL
My_PF70:I.DriveStatus_ActualDir	0	BOOL
My_PF70:I.DriveStatus_Accelerating	0	BOOL
My_PF70:I.DriveStatus_Decelerating	0	BOOL
My_PF70:I.DriveStatus_Alarm	0	BOOL
My_PF70:I.DriveStatus_Faulted	0	BOOL
My_PF70:I.DriveStatus_AtSpeed	0	BOOL
My_PF70:I.DriveStatus_LocalID0	0	BOOL
My_PF70:I.DriveStatus_LocalID1	0	BOOL
My_PF70:I.DriveStatus_LocalID2	0	BOOL
My_PF70:I.DriveStatus_SpdRefID0	0	BOOL
My_PF70:I.DriveStatus_SpdRefID1	0	BOOL
My_PF70:I.DriveStatus_SpdRefID2	0	BOOL
My_PF70:I.DriveStatus_SpdRefID3	0	BOOL
+ My_PF70:I.OutputFreq	0	INT
+ My_PF70:I.AccelTime1	0	INT
+ My_PF70:I.DecelTime1	0	INT
+ My_PF70:I.Fault1Code	0	INT
+ My_PF70:I.DCBusVoltage	0	INT
+ My_PF70:I.OutputCurrent	0	INT
+ My_PF70:I.OutputPower	0	INT

图 25-20　输入标签

已经为已组态的 I/O 生成说明性标签名称。会明确定义“变频器状态”位（BOOL）以及“输出频率”和“数据链路”。请注意为每个标签使用合适的数据类型。

19. 验证输出标签。现在展开 My_ PF70E: O 输出，然后向下滚动，查看输出标签名称，如图 25-21 所示。

已经为已组态的 I/O 生成说明性标签名称。会明确定义“变频器逻辑”位（BOOL）以及“给定频率”和“数据链路”。请注意为每个标签使用合适的数据类型。数据链路 B1、B2、C1 和 C2 未使用，但仍需为其分配标签名称，这样它们的名称才会包含字“undefined”，说明此次未使用该数据链路（未连接到参数）。

关闭“控制器标签”（Controller Tags）窗口。

20. 打开 PowerFlex 70 模块属性。双击 I/O 组态文件夹中的 PowerFlex 70 节点，以显示“模块属性”（Module Properties）窗口，如图 25-22 所示。单击“变频器”（Drive）选项卡。

21. 查看 PowerFlex 70 的“变频器”（Drive）选项卡。“变频器”（Drive）选项卡是版本 16 中的新增内容。之前的 RSLogix 5000 版本包含 DriveExecutive Lite，可以通过变频器的“模块属性”（Module Properties）窗口上的图标启动 DriveExecutive Lite（但仍需在计算机上安装）。变频器组态软件现在与 RSLogix 5000 已完全集成，不再是两个单独的软件，如图 25-23 所示。

Name	Value	Data Type
+ My_PF70:I	{...}	AB:PowerFlex70_EEE2565D:I:0
− My_PF70:O	{...}	AB:PowerFlex70_942C105B:O:0
+ My_PF70:O.DriveLogicRslt	2#0000_0000_...	INT
My_PF70:O.DriveLogicRslt_Stop	0	BOOL
My_PF70:O.DriveLogicRslt_Start	0	BOOL
My_PF70:O.DriveLogicRslt_Jog	0	BOOL
My_PF70:O.DriveLogicRslt_ClearFault	0	BOOL
My_PF70:O.DriveLogicRslt_Forward	0	BOOL
My_PF70:O.DriveLogicRslt_Reverse	0	BOOL
My_PF70:O.DriveLogicRslt_LocalContrl	0	BOOL
My_PF70:O.DriveLogicRslt_MOPInc	0	BOOL
My_PF70:O.DriveLogicRslt_Accel1	0	BOOL
My_PF70:O.DriveLogicRslt_Accel2	0	BOOL
My_PF70:O.DriveLogicRslt_Decel1	0	BOOL
My_PF70:O.DriveLogicRslt_Decel2	0	BOOL
My_PF70:O.DriveLogicRslt_SpdRefID0	0	BOOL
My_PF70:O.DriveLogicRslt_SpdRefID1	0	BOOL
My_PF70:O.DriveLogicRslt_SpdRefID2	0	BOOL
My_PF70:O.DriveLogicRslt_MOPDec	0	BOOL
+ My_PF70:O.CommandedFreq	0	INT
+ My_PF70:O.AccelTime1	0	INT
+ My_PF70:O.DecelTime1	0	INT
+ My_PF70:O.Undefined_B1	0	INT
+ My_PF70:O.Undefined_B2	0	INT
+ My_PF70:O.Undefined_C1	0	INT
+ My_PF70:O.Undefined_C2	0	INT

图 25-21 输出标签

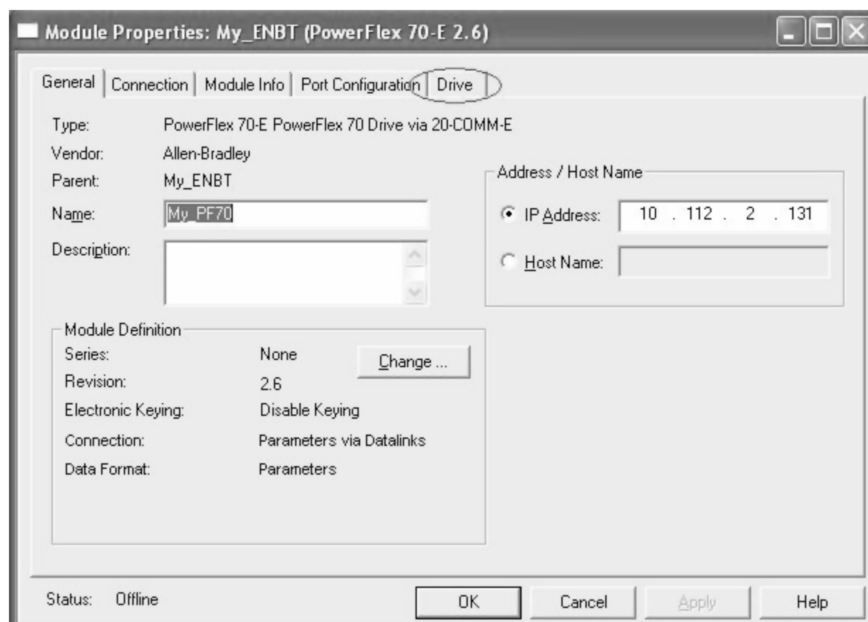


图 25-22 变频器模块属性

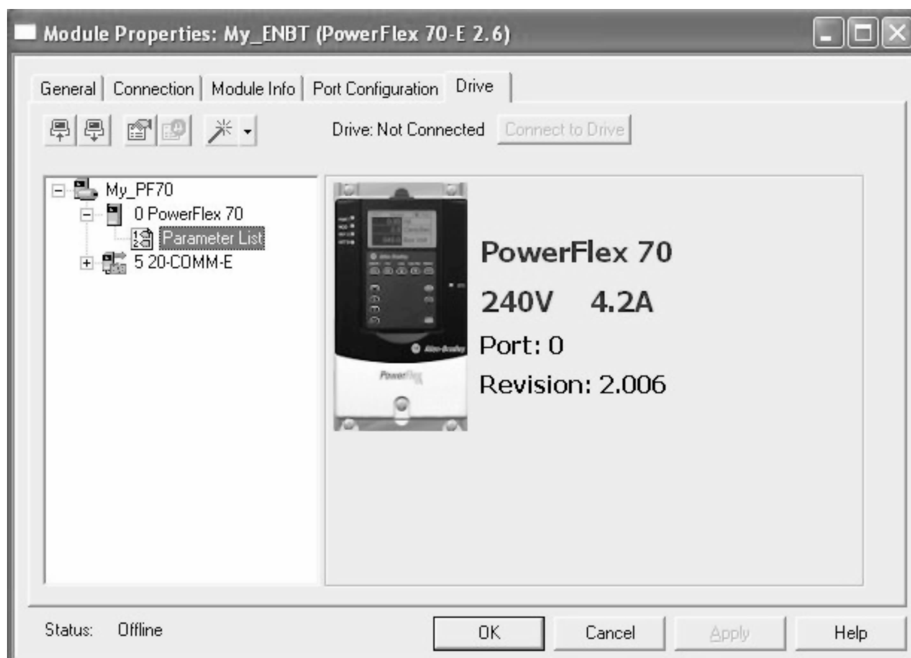


图 25-23　变频器模块属性

DriveExecutive 和 DriveExplorer 的许多常见功能现在都合并到“变频器”（Drive）选项卡中：

1）“菜单”（Menu）树；

2）“状态视图”（Status View）显示界面；

3）图标；

4）向导；

5）“线性列表”（Linear List）编辑器；

6）“文件”（File）/“组”（Group）/“参数”（Parameter）编辑器；

7）报警和事件；

8）“设备属性”（Device Properties）；

9）“诊断项”（Diagnostic Items）。

“变频器”（Drive）选项卡还具有导入/导出功能，因此维护人员或其他没有权限使用 RSLogix 5000 的人员可继续使用 DriveExecutive。

22. 设置 20-COMM-E 的子网组态参数。单击“端口组态”（Port Configuration）并输入“子网掩码”（Subnet Mask）地址，如图 25-24 所示。

输入数值之后，单击“设置”（Set）。IP 地址的组态完成。

23. 查看自动设置的网络参数。再次单击“变频器”（Drive）选项卡。在树形视图中展开 20-COMM-E，然后双击其“参数列表”（Parameter List），查看“线性列表”（Linear List）编辑器。向下滚动，查看参数 23、35 和 36 的值，如图 25-25 所示。如上文所述，在 PowerFlex 70“模块定义”（Module Definition）窗口中组态 I/O 连接时，已对这些参数进行了自动设置。

单击“关闭”（Close）。

24. 查看自动设置的数据链路参数。双击变频器“参数列表”（Parameter List）。从“所有参数”（All Parameters）下拉菜单中展开“通信”（Communication）文件夹，然后单击“数据链路”

（Datalinks），以显示数据链路参数，如图 25-26 所示。

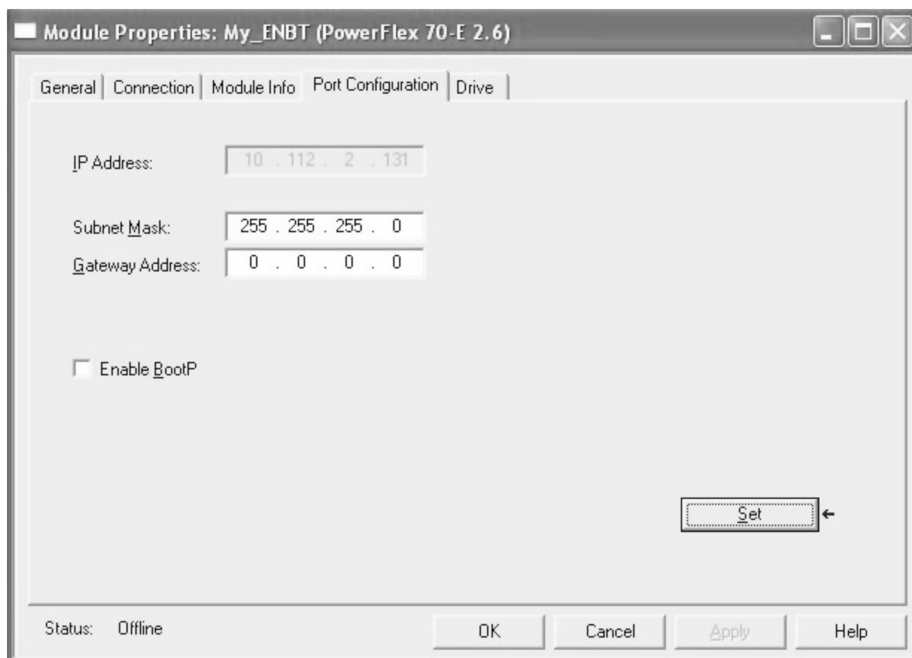
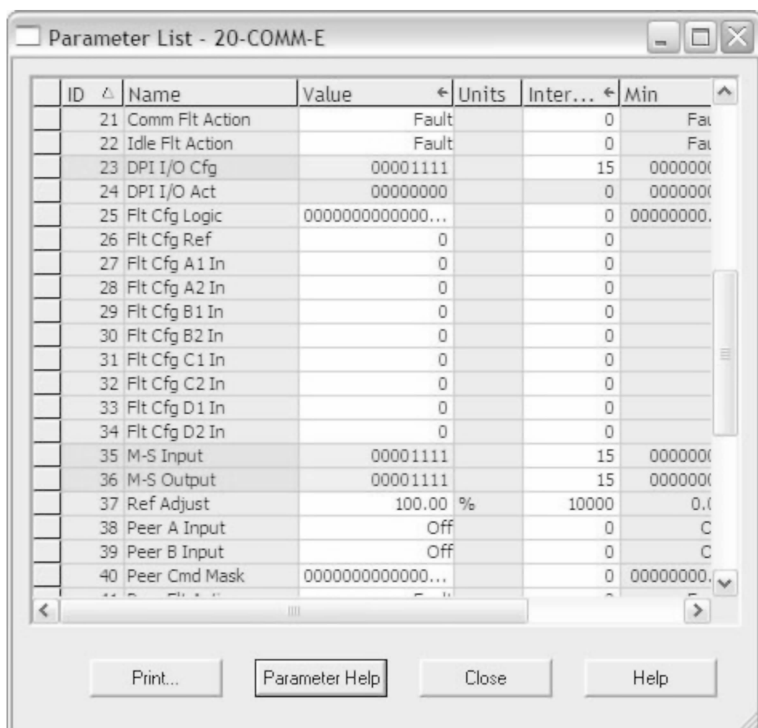


图 25-24　组态端口



ID	Name	Value	Units	Inter...	Min
21	Comm Flt Action	Fault		0	Fau
22	Idle Flt Action	Fault		0	Fau
23	DPI I/O Cfg	00001111		15	000000(
24	DPI I/O Act	00000000		0	000000(
25	Flt Cfg Logic	0000000000000...		0	00000000.
26	Flt Cfg Ref	0		0	
27	Flt Cfg A1 In	0		0	
28	Flt Cfg A2 In	0		0	
29	Flt Cfg B1 In	0		0	
30	Flt Cfg B2 In	0		0	
31	Flt Cfg C1 In	0		0	
32	Flt Cfg C2 In	0		0	
33	Flt Cfg D1 In	0		0	
34	Flt Cfg D2 In	0		0	
35	M-S Input	00001111		15	000000(
36	M-S Output	00001111		15	000000(
37	Ref Adjust	100.00	%	10000	0.(
38	Peer A Input	Off		0	C
39	Peer B Input	Off		0	C
40	Peer Cmd Mask	0000000000000...		0	00000000.

图 25-25　参数列表

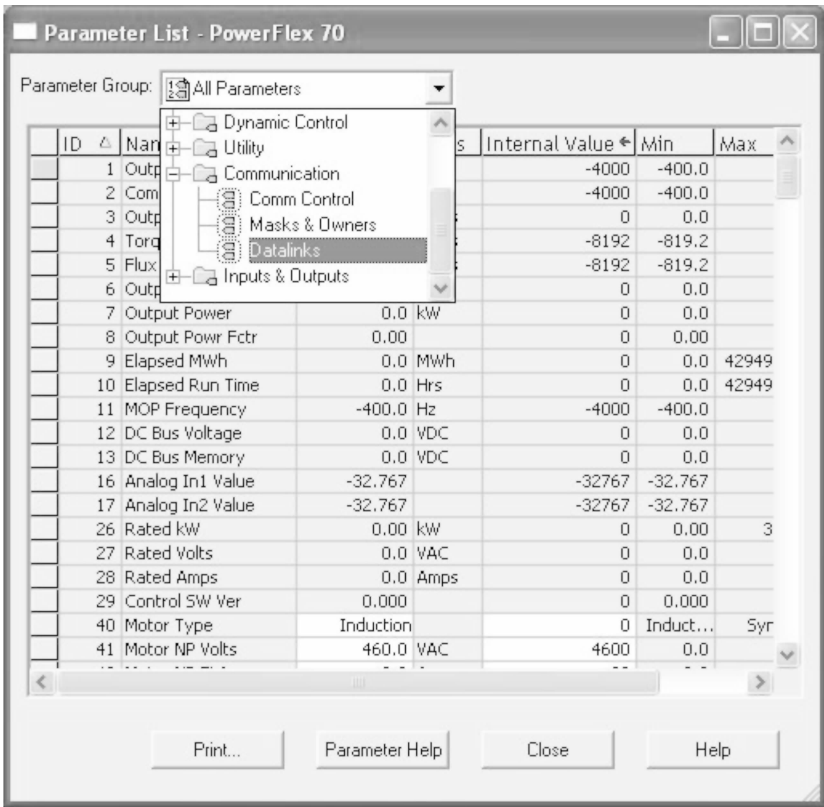


图 25-26　参数列表

25. 验证自动设置的数据链路参数值。查看参数 300、301、310、311、312、313、314 和 315 的值，如图 25-27 所示。如上文所述，在 PowerFlex 70“模块定义”（Module Definition）窗口中组态 I/O 连接时已对这些参数进行了自动设置。

140 是指参数 140［加速时间 1］。

142 是指参数 142［减速时间 1］。

243 是指参数 243［故障代码 1］。

12 是指参数 12［直流母线电压］。

3 是指参数 3［输出电流］。

单击“关闭”（Close）。

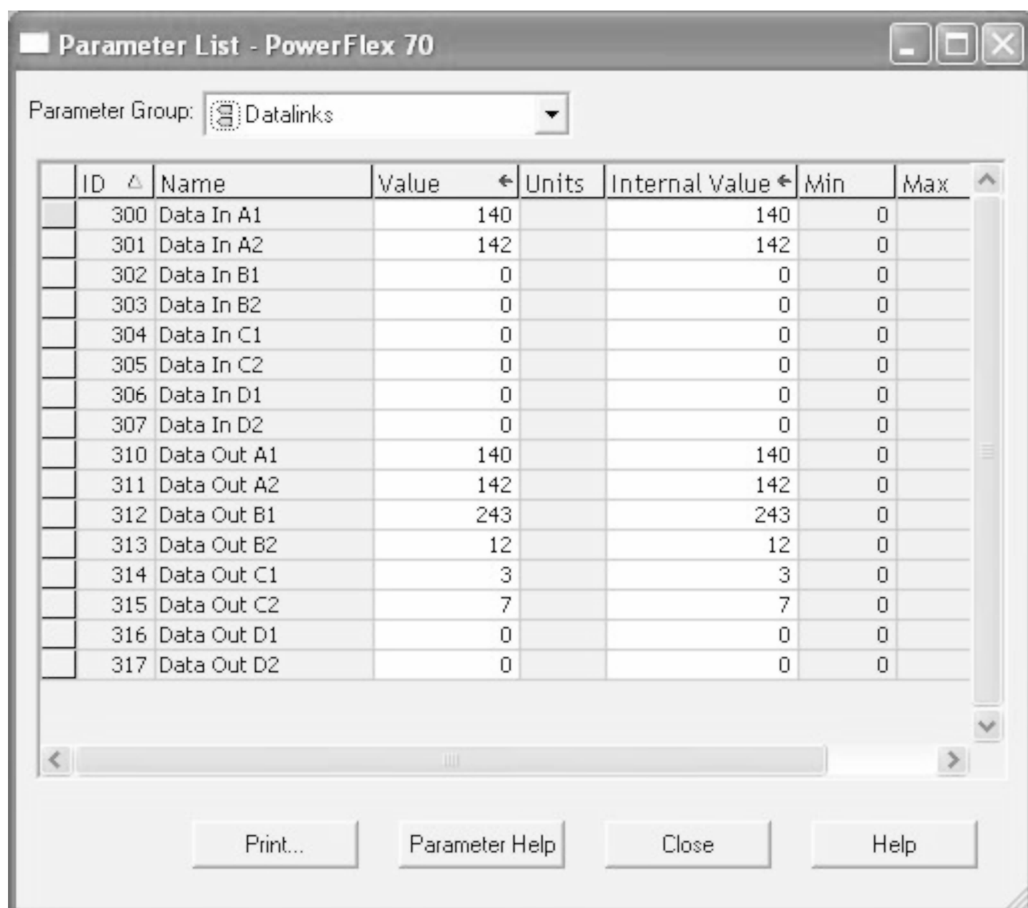
26. 启动 PowerFlex 70 启动向导：单击工具栏中的向导图标，查看可用向导。选择 PowerFlex 70/70 启动向导并单击“选择”（Select），如图 25-28 所示。

27. 查看该向导的警告对话框。将出现“向导启动确认”（Wizard Launch Confirmation）对话框，如图 25-29 所示。该消息会提醒您使用该软件来组态变频器时，如果不当可能会出现意外的、危险的机械运动。

单击“是”（Yes）打开启动向导。

28. 开始执行 PowerFlex 70 启动向导的第 1 步操作，如图 25-30 所示。

启动向导是一种易于使用的工具，可以用它对变频器进行组态。用户不需要知道参数编号，而是使用组态的说明性文本和图形化表示。



ID	Name	Value	Units	Internal Value	Min	Max
300	Data In A1	140		140	0	
301	Data In A2	142		142	0	
302	Data In B1	0		0	0	
303	Data In B2	0		0	0	
304	Data In C1	0		0	0	
305	Data In C2	0		0	0	
306	Data In D1	0		0	0	
307	Data In D2	0		0	0	
310	Data Out A1	140		140	0	
311	Data Out A2	142		142	0	
312	Data Out B1	243		243	0	
313	Data Out B2	12		12	0	
314	Data Out C1	3		3	0	
315	Data Out C2	7		7	0	
316	Data Out D1	0		0	0	
317	Data Out D2	0		0	0	

图 25-27　数据链路参数

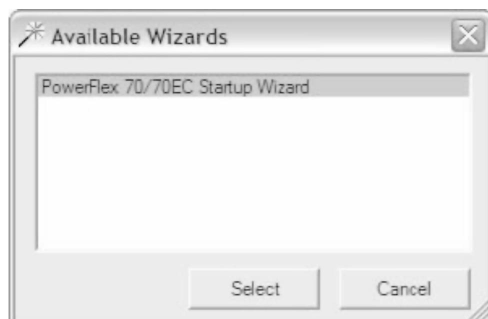


图 25-28　可用向导

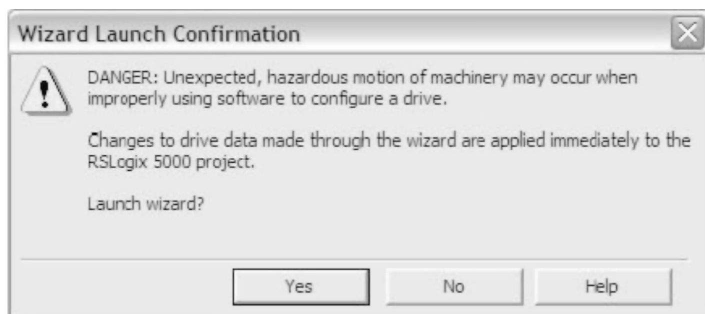


图 25-29　向导启动确认

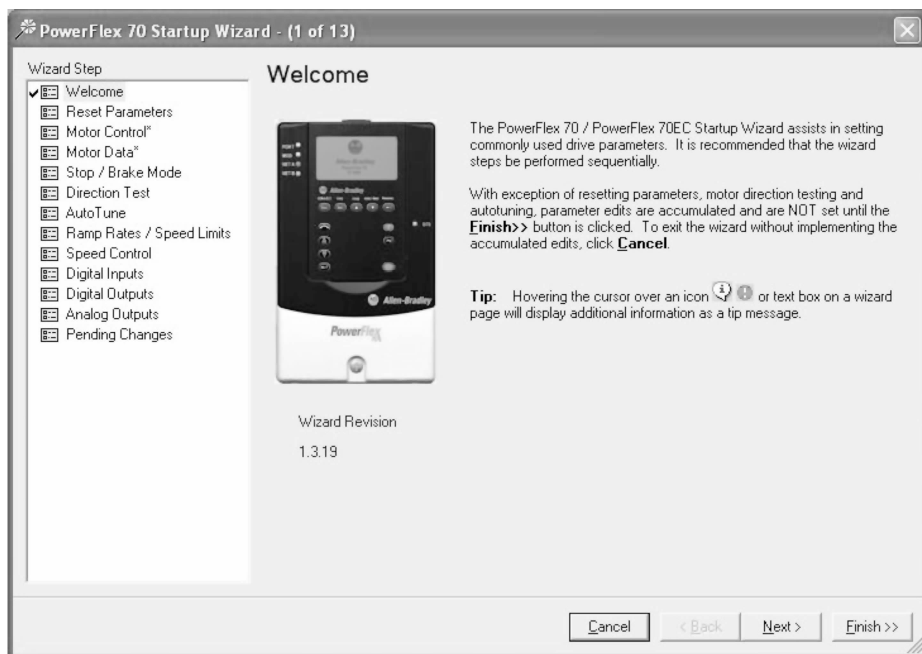


图 25-30　变频器启动向导

单击“下一步 >”（Next >）查看顺序过程，然后继续单击“下一步 >”（Next >），直至进入向导的第 4 步。

29. 执行 PowerFlex 70 启动向导的第 4 步操作，如图 25-31 所示。

将电动机 NP 功率设置为 1，将电动机 NP FLA 设置为 1。

单击“下一步_ >”（Next >），继续查看顺序过程，然后继续单击“下一步 >”（Next >），直至进入向导的第 9 步。

30. 执行 PowerFlex 70 启动向导的第 8 步操作，如图 25-32 所示。

为“最大速度”（Max Speed）输入 90，为“最小速度”（Min Speed）输入 10。为 S 曲线输入非零值，查看图形出现哪些变化。然后将其重置为 0。

单击“下一步”（Next）。

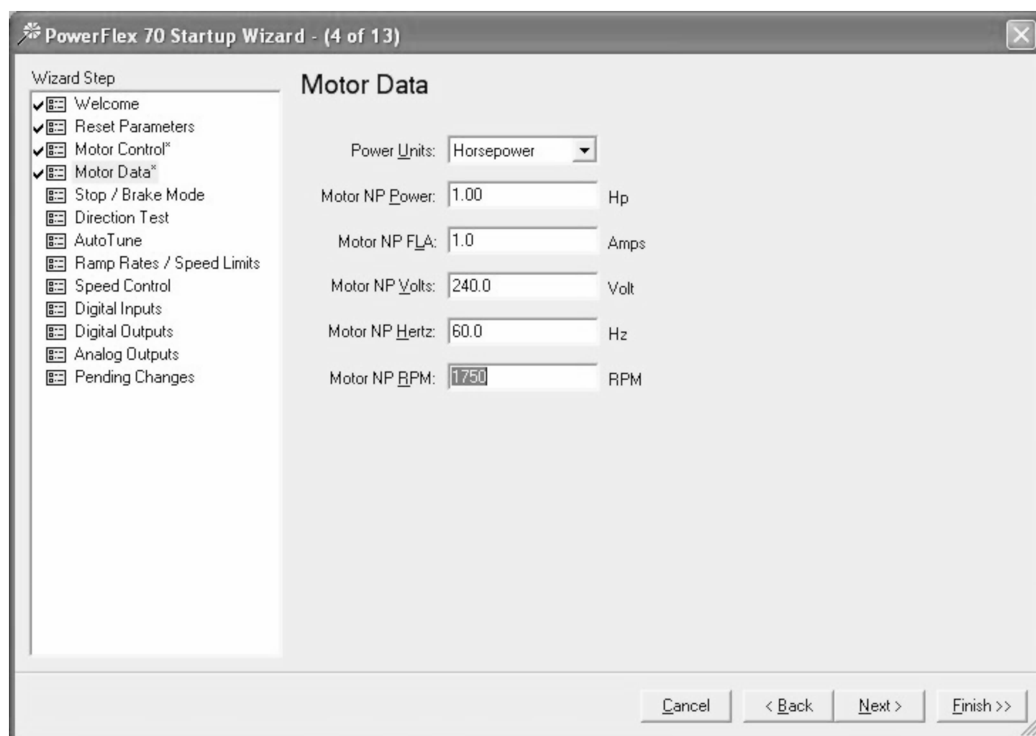


图 25-31　变频器启动向导

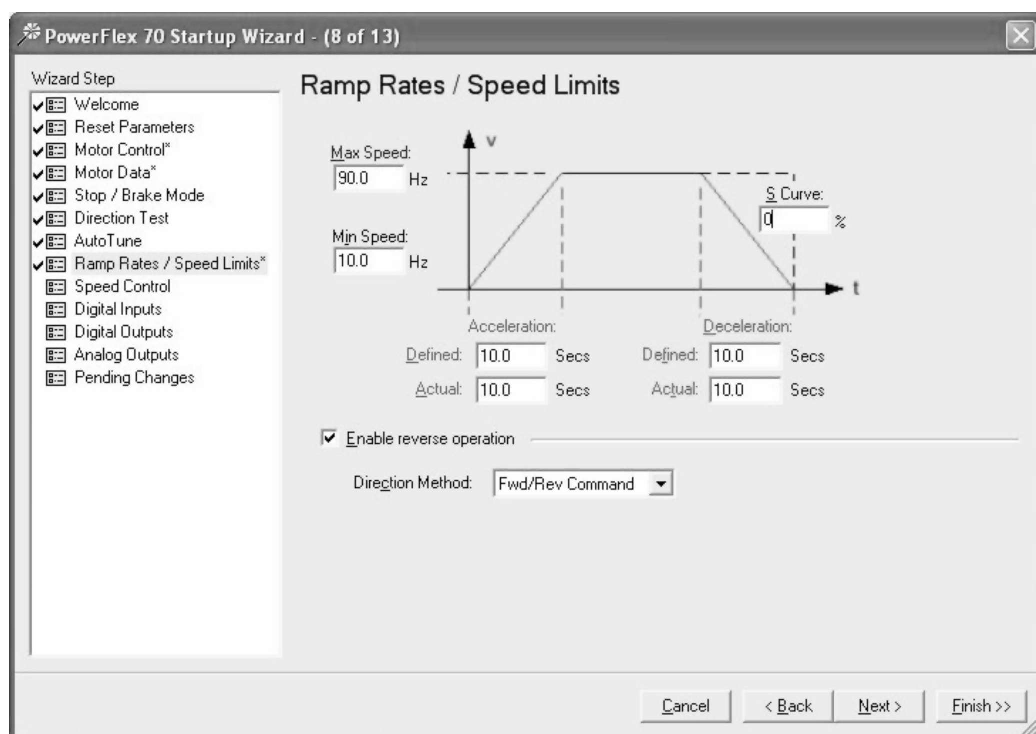


图 25-32　变频器启动向导

31. 执行 PowerFlex 70 EC 启动向导的第 9 步和第 10 步操作，如图 25-33、图 25-34 所示。

单击“速度基准值”（Speed Reference）下拉箭头并选择“DPI 端口 5”（DPI Port 5）。注意图形是如何变化以显示基准值将出现在哪里。20 – COMM – E EtherNet/IP 适配器会插入 DPI 端口 5，我们要通过以太网网络控制 ControlLogix 的基准值。

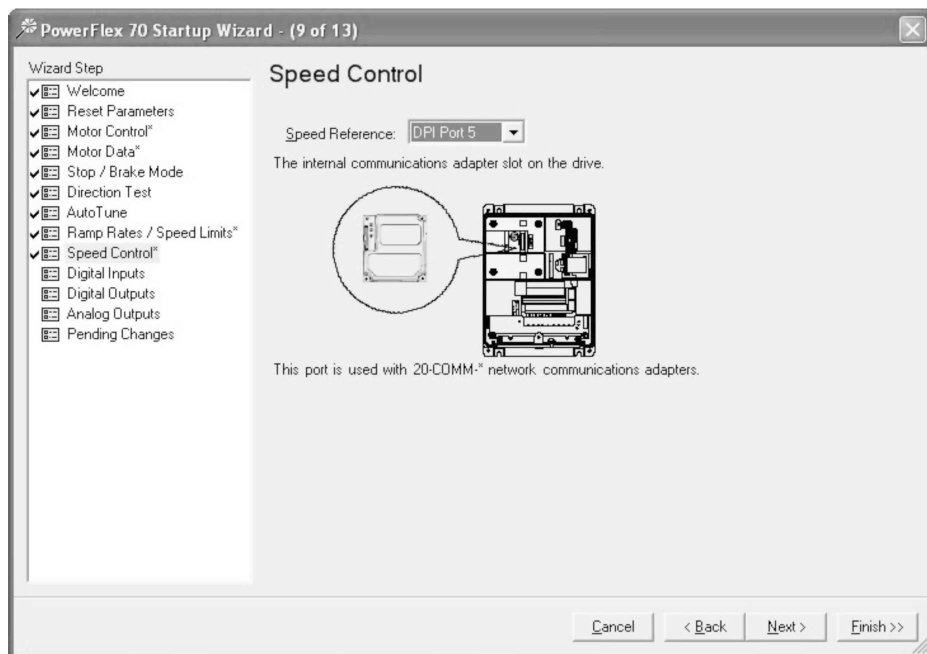


图 25-33　变频器启动向导

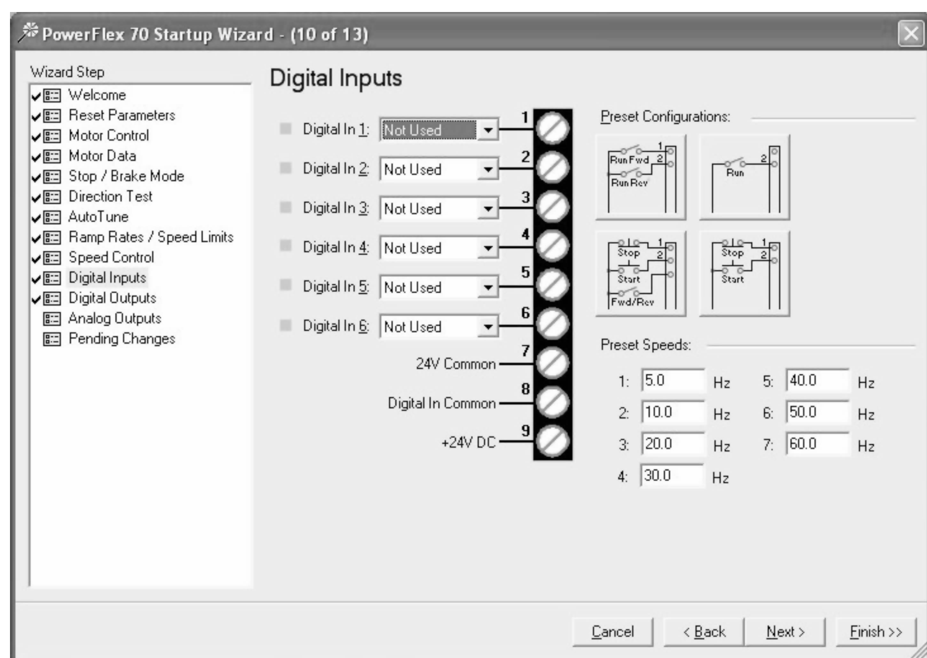


图 25-34　变频器启动向导

继续单击“下一步”（Next >），直至进入向导的第 13 步。

32. 执行 PowerFlex 70 EC 启动向导的第 13 步操作，如图 25-35 所示。所有已应用更改和未决更改都会列出。

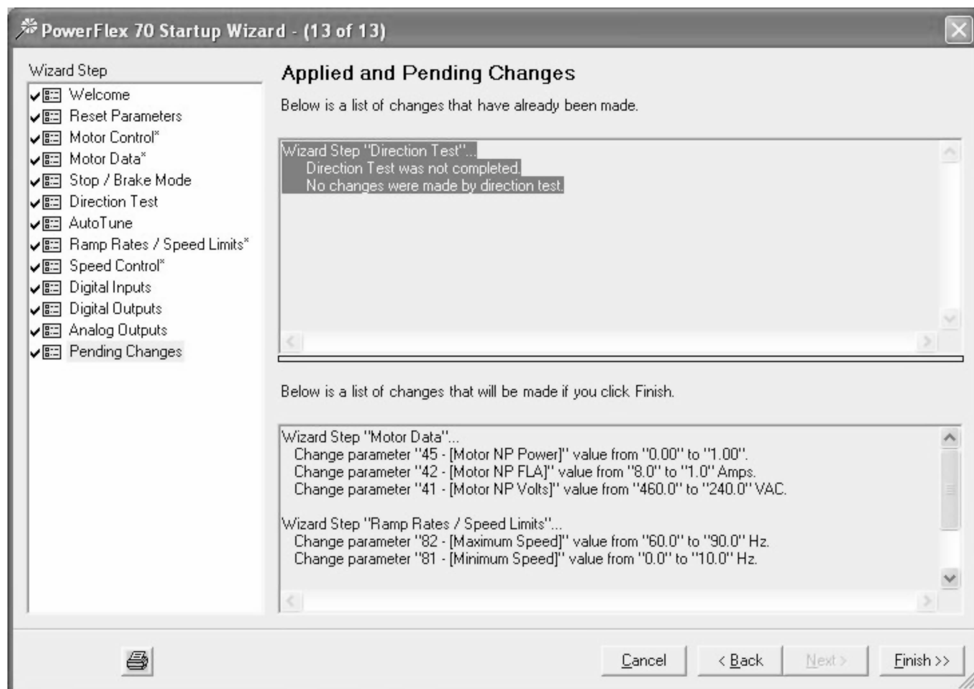


图 25-35 变频器启动向导

单击“完成”（Finish >>），接受这些更改。

可以通过启动向导快速、方便地对变频器进行组态。虽然启动向导并未包括每一参数，但包括了大多数应用的常用参数。

33. 下载变频器组态。目前为止，所执行的操作都是在离线状态下进行的。现在，必须获取此新组态并将其传送至在线变频器。

单击工具栏中的“下载参数数据”图标，启动变频器下载过程。

将显示“连接至变频器”（Connect To Drive）窗口。展开以太网或 Ethernet-IP 驱动器（取决于在计算机上组态的是哪一个），找到 PowerFlex 70 EC 变频器图标并选择该图标。然后单击“确定”（OK）。集成变频器配置文件将尝试连接并检验变频器及其连接的外围设备。这一过程大概需要 10 s。

如果您尝试连接的变频器包含的参数和 I/O 组态与 RSLogix 5000 项目不同，则可能会显示如图 25-36 所示窗口。

出现该窗口之后，单击“使用项目”（Use Project）。项目设置将会下载到变频器。适配器和变频器将执行复位，复位过程大概将持续 60s 才会完成。复位完成之后，请单击随后出现（未呈灰色显示）的“继续”（Continue）。

完成此操作之后，将显示“选择要下载的设备”（Select Devices To Download）窗口（见图 25-37）。出现该窗口后，软件会通知您在离线组态与在线设备（PF70、20 – COMM – E、HIM 等）之间发现设备属性差异（组态、版本等）。

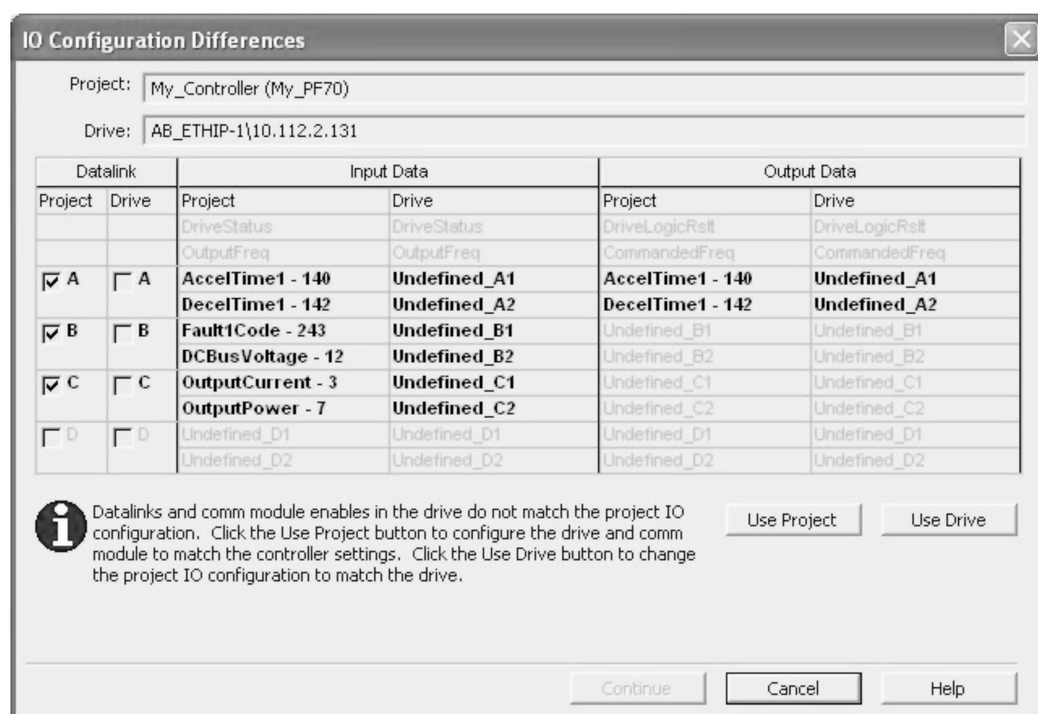


图 25-36　IO 不同配置

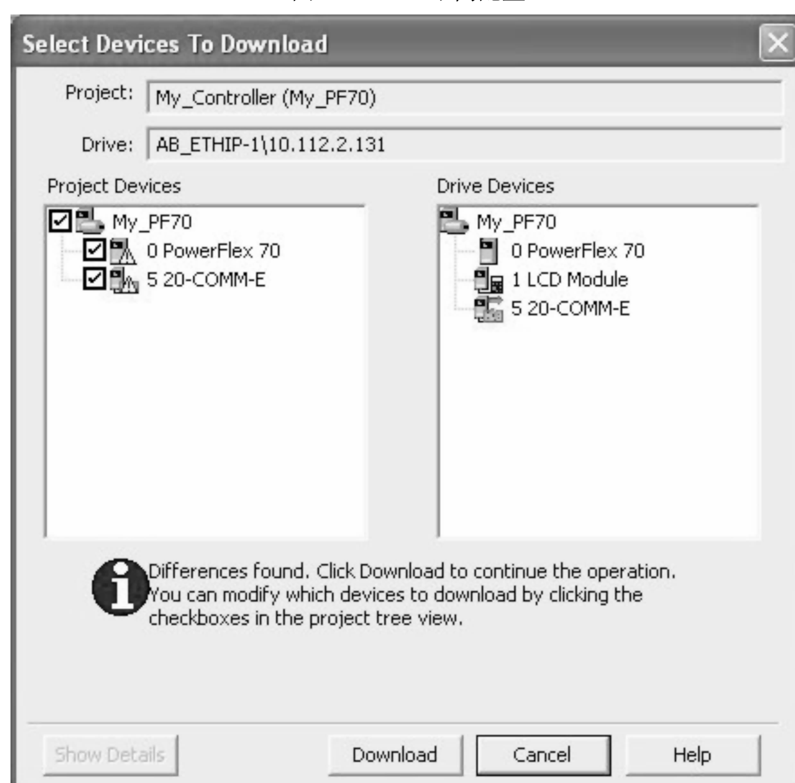


图 25-37　选择要下载的设备

单击“下载”（Download）合并这些新更改。

将显示下载状态框。整个下载过程可能会持续 10s。下载完成之后，下载状态框会消失。

34. 在进入在线状态之前完成变频器设置。下载完成后，会返回“模块属性”（Module Properties）窗口，如图 25-38 所示。

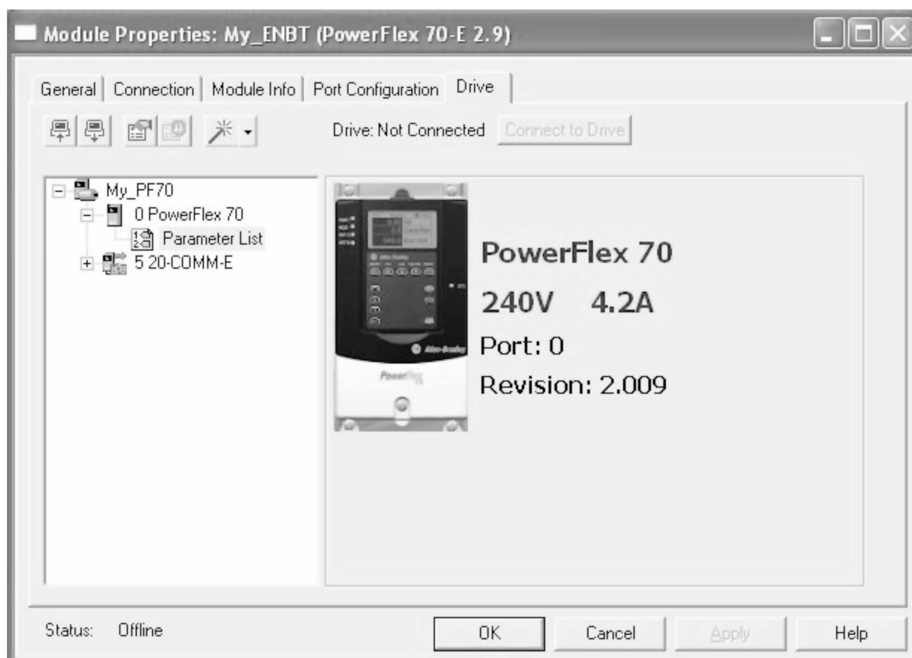


图 25-38 模块属性

单击“确定”（OK）。

三、下载项目并验证网络 I/O 可正常运行

1. 将 RSLogix 5000 项目下载到控制器。

单击（RSWho）图标（或选择“通信 > 在线节点”（Communications > Who Active）），然后展开节点 10.112.2.1x3（x 是表单号），以查看控制器背板。选择插槽 0 中的控制器，单击“设置项目路径”（Set Project Path），然后单击“下载”（Download），如图 25-39 所示。

2. 确认下载。

如果执行该步骤之前控制器处于编程模式，则会出现以下窗口，如图 25-40 所示。单击“下载”（Download），将项目下载到控制器。

然而，如果执行该步骤之前控制器处于运行模式，则会出现以下窗口，如图 25-41 所示。再次单击“下载”（Download），将项目下载到控制器，然后单击“是”（Yes），重新将控制器设为运行模式。

3. 检验控制器的状态。

控制器应处于“编程”模式或“远程编程”模式，I/O OK 指示灯应呈绿色显示，如图 25-42 所示。

I/O 组态树中的 PowerFlex 70 EC 的旁边不应存在黄色三角形。如果连接存在问题，会显示黄色三角形 ⚠，如图 25-43 所示。

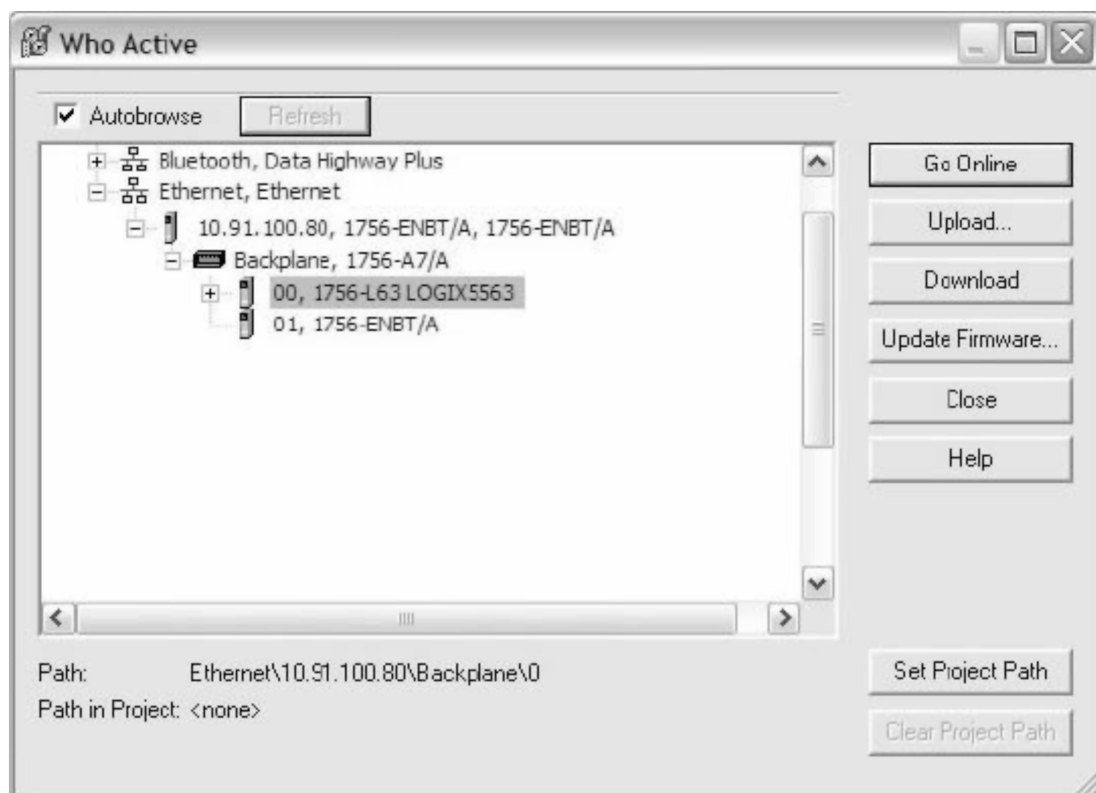


图 25-39　在线节点

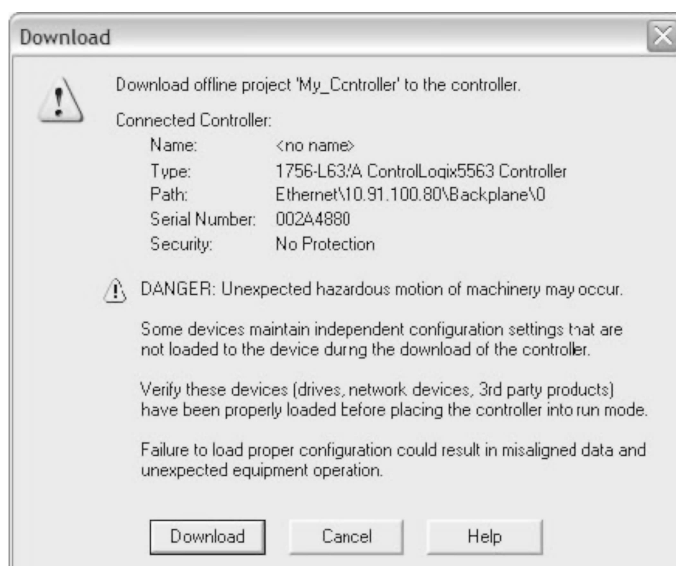


图 25-40　编程模式下载

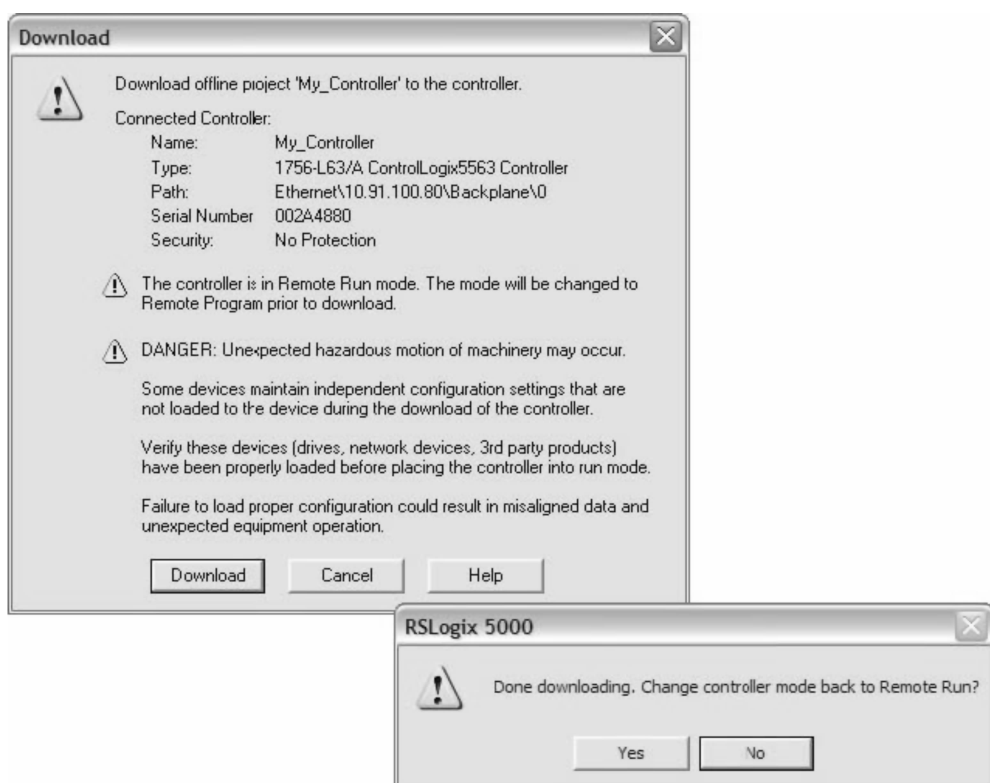


图 25-41　运行模式下载

注：如果这一步骤中显示错误，请联系您的导师。

4. 将控制器设为运行模式。请注意，如果控制器已处于运行模式，则可跳过此步骤。可以单击模式下拉箭头并选择“运行模式”（Run Mode），也可以将控制器开关旋至运行位置，如图 25-44 所示。

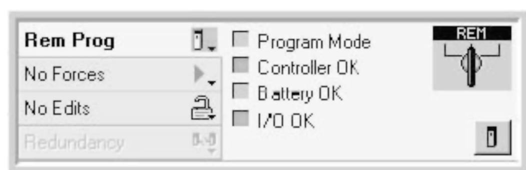


图 25-42　I/O 指示灯

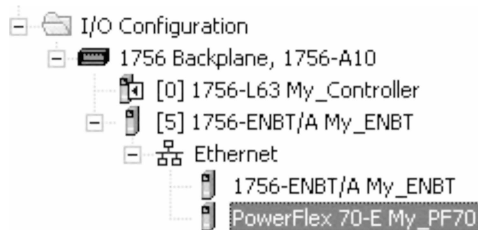


图 25-43　I/O 组态项目列表

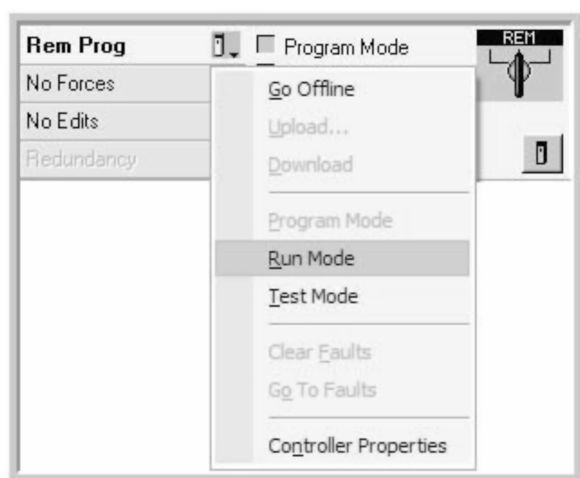


图 25-44　选择运行模式

如果弹出用于确认远程运行的对话框，请单击“是”（Yes），如图 25-45。

5. 查看控制器的运行状态。下列指示灯应呈绿色显示，且控制器应处于远程运行模式或运行模式，如图 25-46 所示。电池 OK 指示灯可能会呈红色显示；对于本实验，表示电池正常工作。

注：如果这一步骤中显示错误，请联系您的导师。

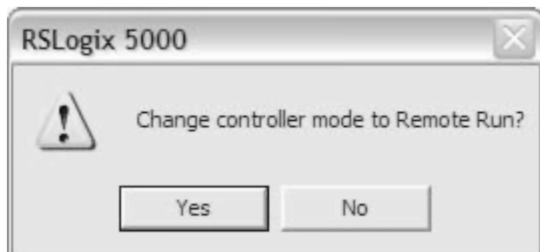


图 25-45　确认对话框

6. 查看控制器的输出标签。双击“菜单”（Menu）树中的“控制器标签”（Controller Tags），然后展开 PowerFlex 70 EC 输出。



图 25-46　控制器状态显示

7. 控制变频器。

输入 AccelTime1 和 DecelTime1 的值。需要注意的是，这些值是以十分之一的倍数表示的，因此，值 20 等于 2.0s。

为 CommandedFreq 输入 0 到 32767 之间的值。该值是以工程单位表示的基准值，32767 是最大频率（90Hz）。

为 ClearFault 输入 1，然后再将其设为 0。这样可以复位变频器中的任何故障。

为 Start 输入 1，然后再将其设置为 0。所有启动都需要上升沿（0 到 1）转换，如图 25-47 所示。

如果需要，可使用 HIM 查看参数 140［加速时间 1］和参数 142［减速时间 1］，以验证控制器是否正在发送数据。

8. 查看控制器的输入标签。展开控制器输入，如图 25-48 所示。

默认情况下，PowerFlex 70 EC 变频器上的基准值采用工程单位，其中：

两者之间是线性关系，如图 25-49 所示，因此可以使用以下表达式：

基准值（以工程单位表示）= 参数 82［最大速度］×32，767/参数 55［最大频率］

对于本实验，我们将参数 82［最大速度］设置为 90.0Hz（变频器默认值为 60.0Hz），将参数 81［最小速度］设置为 10.0Hz（变频器默认值为 10.0Hz）。执行此操作后会生成大小为 2521（10.0Hz）的初始工程单位死区，最大死区约为 22685（90.0Hz）。

您最大可输入大小为 32767 的值，然而在这种情况下，您会得到差值 32767 - 22685，即 10082。如您所见，此类修改可提高灵活性、实现更多分辨率并增大对速度范围的控制。无论是哪种情况，输入的值都会使用同一条线性曲线。

Feedback 字的作用与 Reference 相似。默认情况下，PowerFlex 70 EC 变频器上的 Feedback 是采用工程单位表示的，其中，32，767 等于参数 55［最大频率］（默认值 = 130Hz），0 等于 0Hz。两者之间是线性关系，因此可以使用以下表达式：

Feedback（以工程单位表示）= 参数 82［最大速度］×32，767/参数 55［最大频率］

	Name	Value	Data Type
	+ My_PF70:I	{...}	AB:PowerF
	− My_PF70:O	{...}	AB:PowerF
	+ My_PF70:O.DriveLogicRslt	2#0000_0000_0000_0000	INT
	My_PF70:O.DriveLogicRslt_Stop	0	BOOL
	My_PF70:O.DriveLogicRslt_Start	0	BOOL
	My_PF70:O.DriveLogicRslt_Jog	0	BOOL
	My_PF70:O.DriveLogicRslt_ClearFault	0	BOOL
	My_PF70:O.DriveLogicRslt_Forward	0	BOOL
	My_PF70:O.DriveLogicRslt_Reverse	0	BOOL
	My_PF70:O.DriveLogicRslt_LocalContrl	0	BOOL
	My_PF70:O.DriveLogicRslt_MOPInc	0	BOOL
	My_PF70:O.DriveLogicRslt_Accel1	0	BOOL
	My_PF70:O.DriveLogicRslt_Accel2	0	BOOL
	My_PF70:O.DriveLogicRslt_Decel1	0	BOOL
	My_PF70:O.DriveLogicRslt_Decel2	0	BOOL
	My_PF70:O.DriveLogicRslt_SpdRefID0	0	BOOL
	My_PF70:O.DriveLogicRslt_SpdRefID1	0	BOOL
	My_PF70:O.DriveLogicRslt_SpdRefID2	0	BOOL
	My_PF70:O.DriveLogicRslt_MOPDec	0	BOOL
	+ My_PF70:O.CommandedFreq	1000	INT
	+ My_PF70:O.AccelTime1	20	INT
	+ My_PF70:O.DecelTime1	20	INT
	+ My_PF70:O.Undefined_B1	0	INT
	+ My_PF70:O.Undefined_B2	0	INT
	+ My_PF70:O.Undefined_C1	0	INT
	+ My_PF70:O.Undefined_C2	0	INT

图 25-47　变频器参数

该值不应该是相同的值。由于将最大速度与最小速度设置为非出厂默认设置，因此，“…CommandedFreq”标签与“…OutputFreq”标签不匹配，并且将变频器设置值应用于该等式时，会得到：

22685 = 90 × 32，767/130

请注意，还会为 OutputCurrent 和 DCBusVoltage 更新数据。

9. 停止变频器。返回变频器输出标签。为 Stop 输入 1，然后再将其设置为 0。现在，您的系统已设置完毕。所有标签均可正常工作！至此，使用新集成变频器配置文件来运行变频器的操作即已结束。

Name	Value	Data Type
− My_PF70:I	{...}	AB:PowerFlex70_EI
+ My_PF70:I.DriveStatus	2#0000_1111_0000_1111	INT
My_PF70:I.DriveStatus_Ready	1	BOOL
My_PF70:I.DriveStatus_Active	1	BOOL
My_PF70:I.DriveStatus_CommandDir	1	BOOL
My_PF70:I.DriveStatus_ActualDir	1	BOOL
My_PF70:I.DriveStatus_Accelerating	0	BOOL
My_PF70:I.DriveStatus_Decelerating	0	BOOL
My_PF70:I.DriveStatus_Alarm	0	BOOL
My_PF70:I.DriveStatus_Faulted	0	BOOL
My_PF70:I.DriveStatus_AtSpeed	1	BOOL
My_PF70:I.DriveStatus_LocalID0	1	BOOL
My_PF70:I.DriveStatus_LocalID1	1	BOOL
My_PF70:I.DriveStatus_LocalID2	1	BOOL
My_PF70:I.DriveStatus_SpdRefID0	0	BOOL
My_PF70:I.DriveStatus_SpdRefID1	0	BOOL
My_PF70:I.DriveStatus_SpdRefID2	0	BOOL
My_PF70:I.DriveStatus_SpdRefID3	0	BOOL
+ My_PF70:I.OutputFreq	1000	INT
+ My_PF70:I.AccelTime1	20	INT
+ My_PF70:I.DecelTime1	20	INT
+ My_PF70:I.Fault1Code	85	INT
+ My_PF70:I.DCBusVoltage	3091	INT
+ My_PF70:I.OutputCurrent	0	INT
+ My_PF70:I.OutputPower	0	INT

图 25-48　控制器输入标签

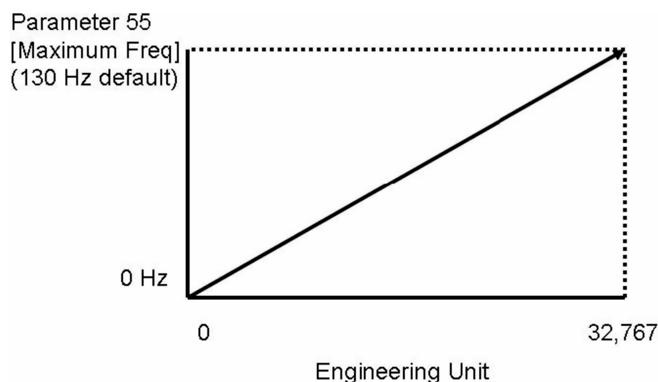


图 25-49　频率与工程单位的线性曲线图

第七部分　集成架构构建器

实验 26：IAB 基础知识动手实验

一、实验目的

1）了解 IAB 用户界面的使用方法；

2）使用子系统向导来添加控制器、网络以及本地 I/O 和分布式 I/O；

3）创建网络并通过拖放的方式添加网络节点；

4）使用 IAB 架构视图；

5）生成项目 BOM（物料清单）；

6）生成项目报表。

二、IAB 用户界面

1. 如果 IAB 已在运行，请跳至步骤 3。

如果尚未运行 IAB，请双击计算机桌面上的“集成架构构建器”（Integrated Architecture Builder）图标，以启动 IAB。

如果 IAB 询问：是否要建立 Internet 连接，请单击“否”（No），如图 26-1 所示。

如果选择“是”（Yes），IAB 将连接到 Internet，以下载 IAB 数据库、软件和文档的可用更新。可通过选择“文件 > Internet 连接设置”（File > Internet Connection Settings），来选择 IAB 在启动时，自动连接到 Internet、从不连接到 Internet，还是在每次启动时都询问：是否连接。

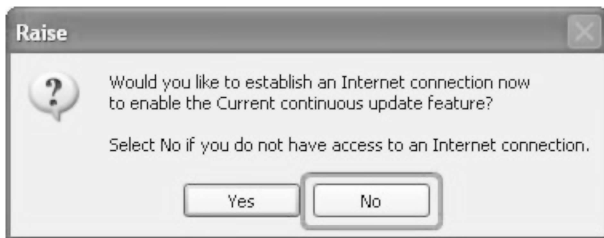


图 26-1　确认 Internet 连接

将出现 IAB 起始页，其中会显示“IAB 项目选项”（IAB Project Options），如图 26-2 所示。

执行下一操作之前，请花费 1min 的时间来了解起始页上的选项。

“快速入门”（Quick Start）选项卡中包括：

1）IAB 中的新功能（What’s New in IAB）：列出了当前版本和几个先前版本的新功能。

2）IAB 项目选项（IAB Project Options）：可在此处启动新项目或打开现有项目。

3）关于 IAB（About IAB）：一些关于 IAB 的信息，例如当前版本和系统要求。

“资源中心”（Resource Center）选项卡中包括：

应用文档（Application Documents）：简要快速入门和深入实验，可帮助了解 IAB 先返回“IAB 项目选项”（IAB Project Options）页面，然后继续进行操作。

2. “IAB 项目选项”（IAB Project Options）页面可以提供以下选项：

1）新建项目（New Project）：创建并打开新的 IAB 工作区。可选择要创建哪种类型的工作区（请参见步骤 3）。

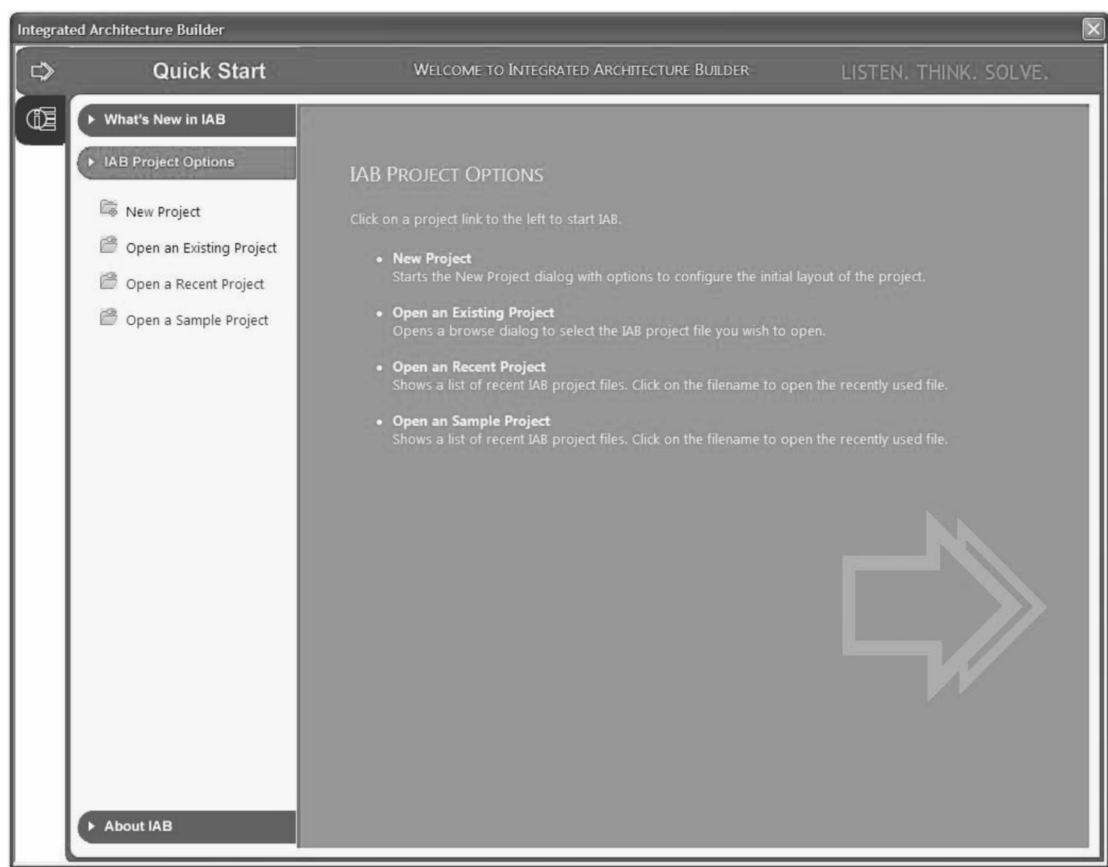


图 26-2　IAB 项目选项图

2）打开现有项目（Open an Existing project）：可以浏览已存储的 IAB 项目文件（可以将 IAB 项目存储在任意目录下。默认情况下，IAB 会将项目存储在 C：\ Program Files \ PST \ RA_IAB 下。每个项目的文件都会保存在名称为项目名称的文件夹中）。

3）打开最近的项目（Open a Recent Project）：显示最近操作过的项目的列表。单击要打开的项目的链接。

4）打开示例项目（Open a Sample Project）：显示示例 IAB 项目的列表，可以打开这些示例 IAB 项目来查看如何使用 IAB。单击要打开的项目的链接。

单击“新建项目”（New Project）并跳至步骤 4，如图 26-3 所示。



图 26-3　新建项目界面

3. 如果 IAB 已在运行，则应保存任何当前已打开的项目，然后单击“新建”（New）按钮，如图 26-4 所示。

将出现“创建新工作区”（Create a New Workspace）对话框，如图 26-5 所示。

4. 在“创建新工作区”（Create a New Workspace）对话框中，选择“空白工作区”（Blank Workspace），然后

键入工作区名称“IAB Basics Lab”，然后单击“确定”（OK），如图 26-6 所示。

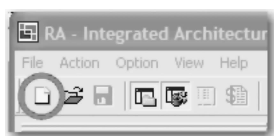


图 26-4　新建按钮界面

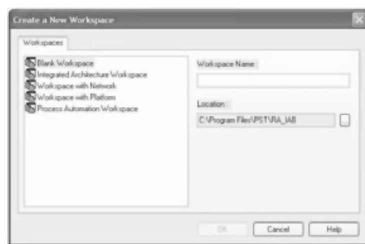


图 26-5　创建新工作区界面

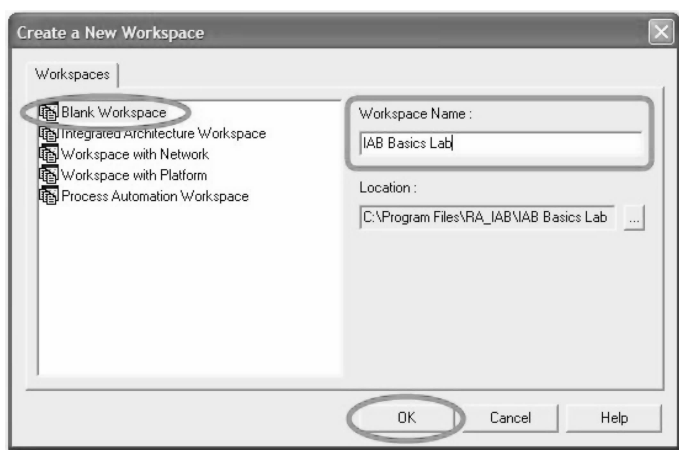


图 26-6　创建新工作区

“创建新工作区”（Create a New Workspace）对话框允许创建多种类型的 IAB 工作区（项目）：

1）空白工作区（Blank workspace）：不包含任何自动创建内容的空白工作区。

2）集成架构工作区（Integrated Architecture Workspace）：自动创建了一个新系统并自动启动了所选择的子系统向导的新工作区。

3）带有网络的工作区（Workspace with Network）：“新网络”（New Network）对话框会被自动打开的新工作区。

4）带有平台的工作区（Workspace with Platform）：“新硬件”（New Hardware）对话框会被自动打开的新工作区。

5）过程自动化工作区（Process Automation Workspace）：将打开用于组态过程控制系统的过程系统估计器的新工作区。

除了过程自动化工作区之外，这些工作区全部相同（自动添加的不同项目除外）。

将打开新的 IAB 工作区。如有必要，请单击“显示/隐藏设备列表”（Show/Hide Device List）按钮来显示其余 IAB 界面，如图 26-7 所示。

5. 花费几分钟的时间来熟悉 IAB 工作区，如图 26-8 所示。

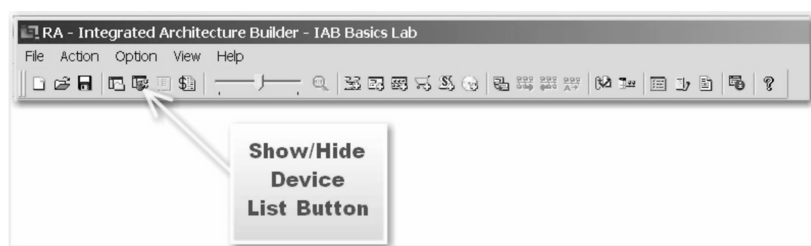


图 26-7　显示/隐藏设备列表

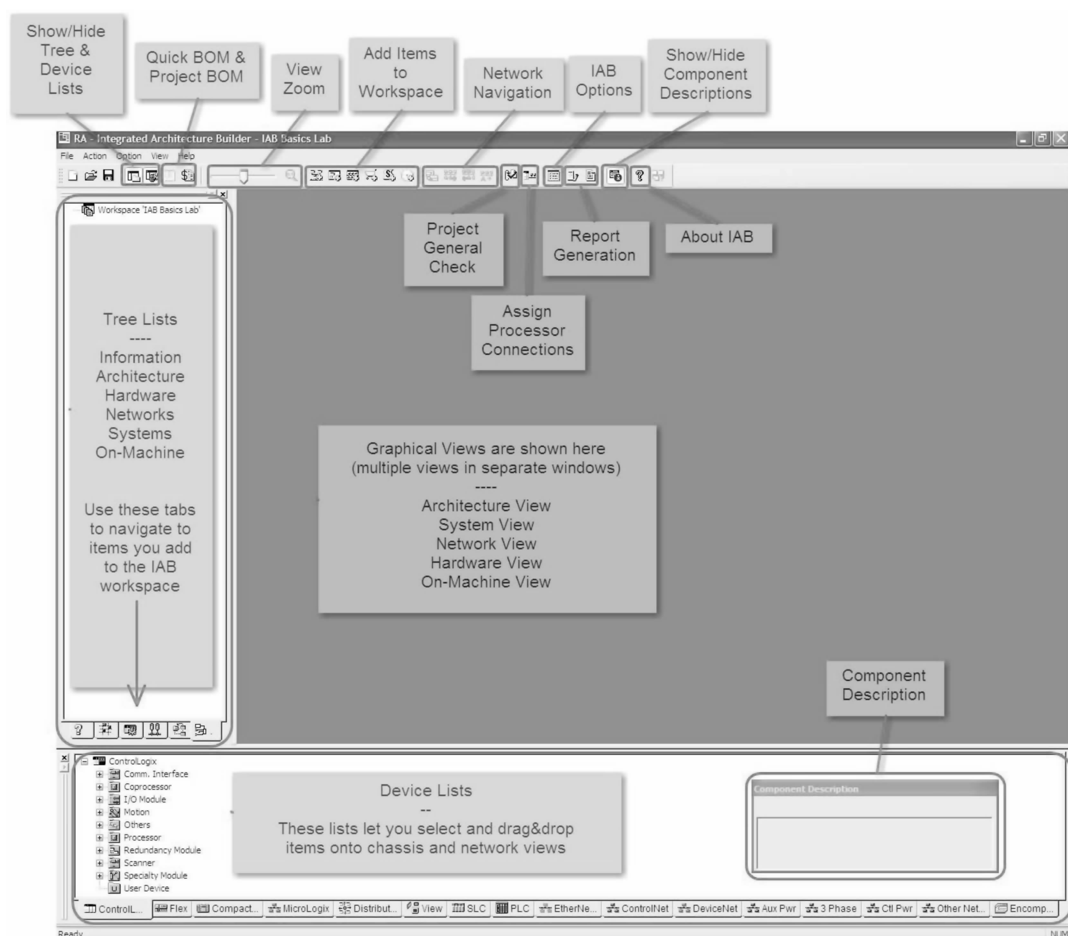


图 26-8　IAB 工作区界面

共有五种类型的图形视图：

1）架构视图：显示已组态的所有内容的逻辑视图，包括网络、硬件和子系统。该视图会在将项目添加到工作区时自动创建。可通过将全局视图中的项目移动到视图中，以及使用其周围的标签创建边界框的方式，对其进行分组。

2）系统视图：系统视图是唯一能够启动子系统向导以添加子系统的位置（稍后会介绍更多有关向导的信息）。

3）网络视图：网络视图会显示项目中的网络 – 每个网络使用一个选项卡，而以太网交换机、ControlNet 网段或 DeviceNet 分支使用子选项卡。每个网络的视图会显示该网络的节点。对于 DeviceNet、ControlNet、EtherNet/IP 和电源网络（辅助、控制和三相），网络视图还会显示网络物理介质。

4）硬件视图：已添加到 IAB 中的项目的每个机架或设备都具有硬件视图。硬件视图会显示机架以及设备的模块。IAB 中的许多组态选项是通过右键单击硬件视图启动的。

5）机旁视图：机旁视图会显示为 I/O 模块组态的机旁布线解决方案。以图形的形式表示接线系统、分线盒和终端设备布线。

接下来，将通过子系统向导添加控制器、网络和 I/O

三、子系统向导

1. 单击“新建系统”（New System）按钮，创建新系统视图，如图 26-9 所示。



图 26-9 新建系统图

将显示新的空白系统视图以及系统工具栏，如图 26-10 所示。

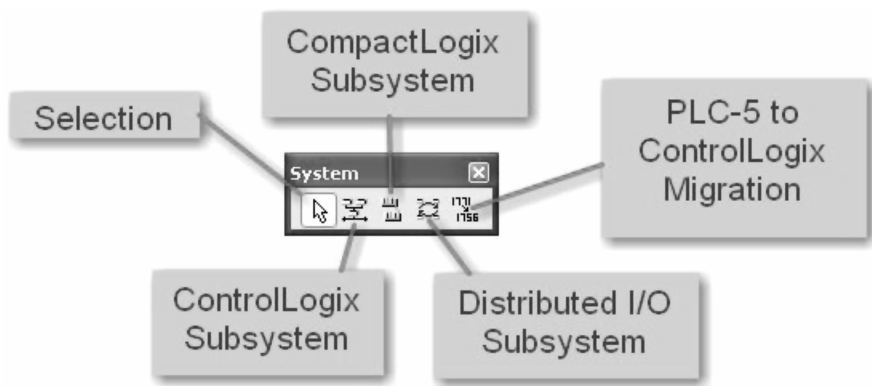


图 26-10 系统工具栏图

系统视图功能

系统视图是将子系统添加到组态以及访问子系统向导的唯一方法。

多系统

可以向 IAB 工作区添加多个系统视图，每个系统视图都可以包含多个子系统。可以通过单击“系统”（Systems）选项卡来查看工作区中的系统列表，以及其中包含的项目。可以通过单击“系统”（System）选项卡，然后单击列表中的系统名称，来随时导航至任何系统视图，如图 26-11 所示。

2. 单击“系统”（System）工具栏中的“ControlLogix 子系统”（ControlLogix Subsystem）按钮，然后将光标移动到“系统视图”（System View）窗口中，并单击鼠标左键。将会在单击的位置出现 CompactLogix 子系统图标（见图 26-12）。如果愿意，可以将子系统图标拖动到系统视图中的其他位置。

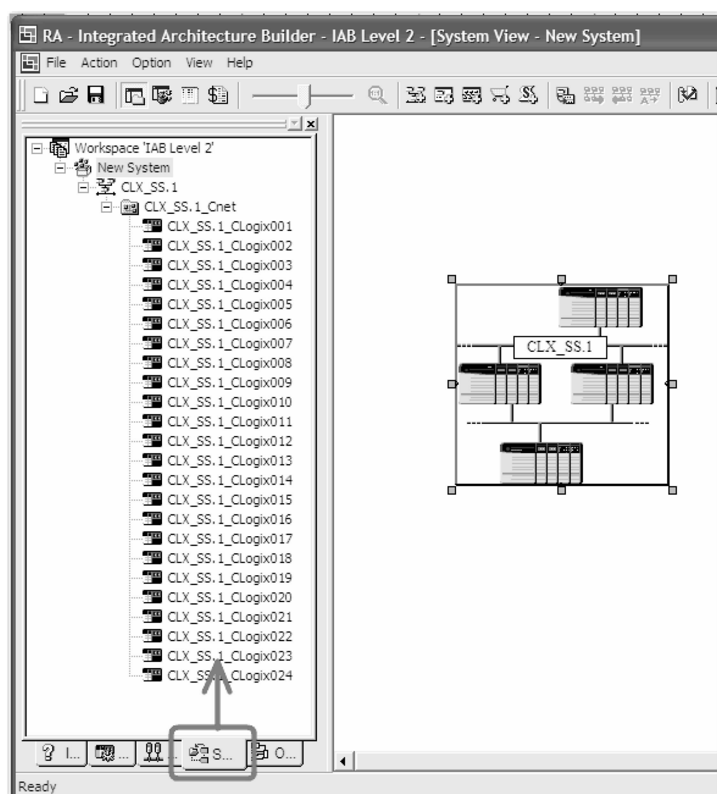


图 26-11　系统视图

3. 在 CLX_SS. 1 子系统名称框中双击，键入名称“ControlLogix 1”（见图 26-13），然后单击“回车”（Enter）（请确保是在名称文本框而非子系统图标中进行单击）。

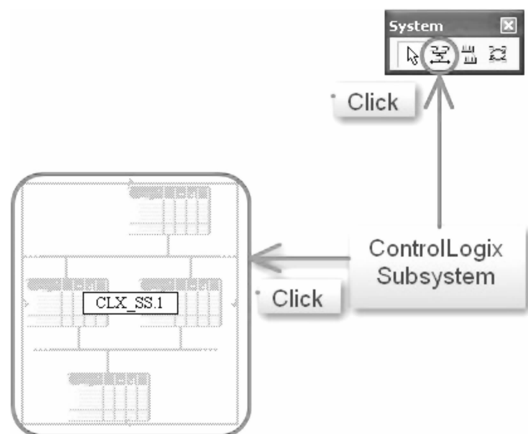


图 26-12　子系统图标

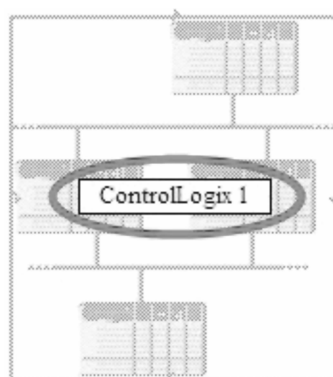


图 26-13　子系统名称框图

4. 打开子系统向导。右键单击子系统图标，然后选择“ControlLogix 子系统向导”（ControlLogix Subsystem Wizard），如图 26-14 所示（还可以通过双击子系统图标来打开该向导）。

ControlLogix 子系统向导将会打开。打开的页面是“机架首选项与 ControlLogix I/O 网络”（Chassis Preferences & ControlLogix I/O Network）页面，如图 26-15 所示。

虽然各子系统向导的具体内容可能会有所不同，但是所有子系统向导都具有这种形式。

5. 要使用 ControlLogix 向导，需要了解向导将会创建的内容。

ControlLogix 向导可以创建以下几项：

1）ControlLogix 机架，该机架包含所指定的 ControlLogix 处理器、通信模块、运动控制模块、本地 ControlLogix I/O，以及在向导中为 ControlLogix I/O 指定的网络。如果要求执行此操作或必须提供指定的 I/O 数量，IAB 会根据需要添加所需通信模块、将其连接到指定网络、然后添加 ControlLogix 机架和 I/O 模块。例如，以下是由 ControlLogix 向导创建的典型网络和机架，如图 26-16 所示。

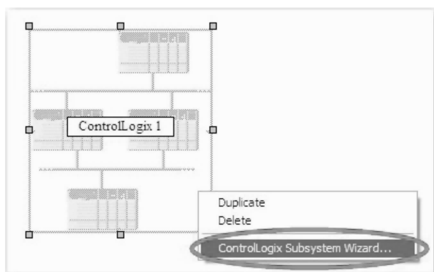


图 26-14　子系统向导图

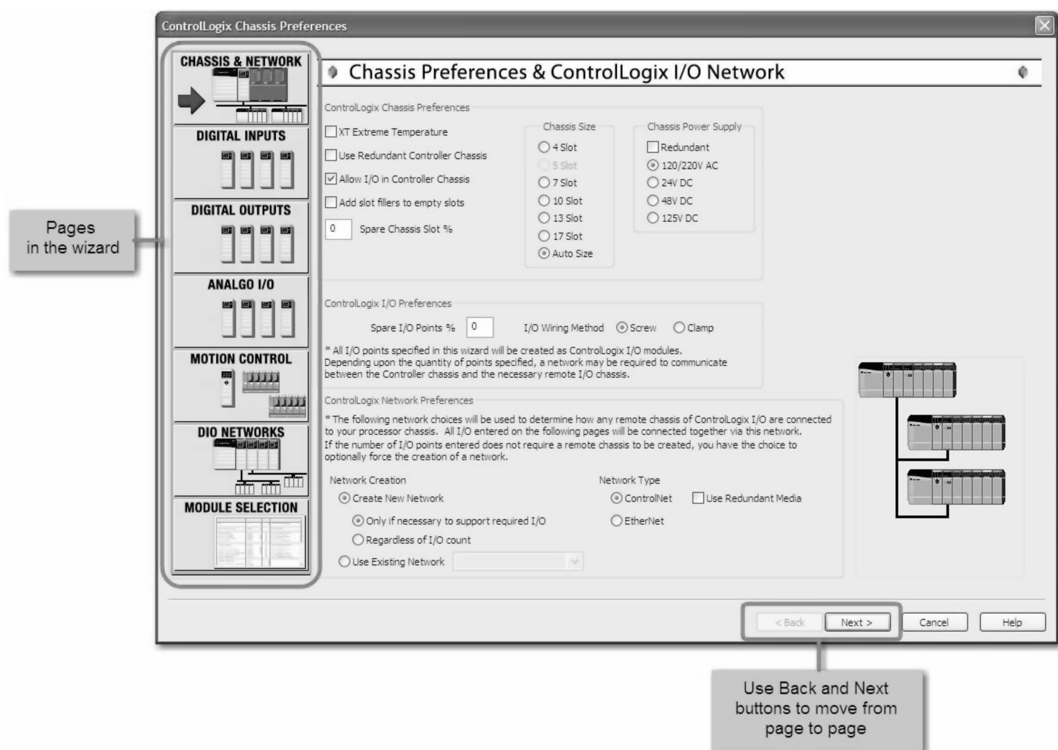


图 26-15　机架首选项

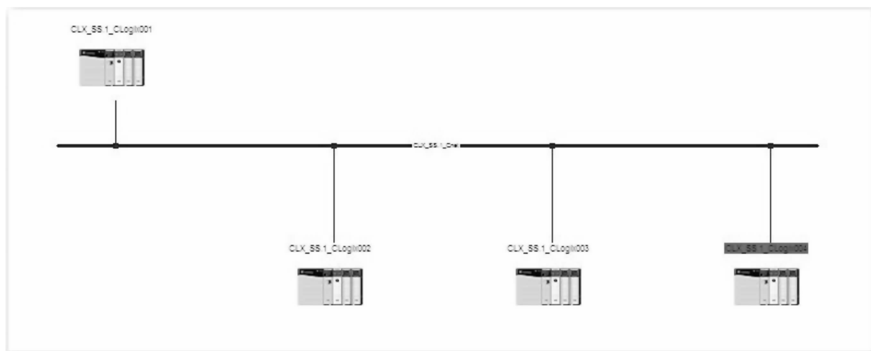


图 26-16　网络和机架图

2）最多为分布式 I/O 指定两个附加网络。在 ControlLogix 子系统向导中，除 ControlLogix I/O 网络之外，还可指定多达两个网络。这些附加的网络用于通过使用 DIO 子系统向导添加的分布式 I/O。例如，如果要将一些 FLEX I/O 模块或 POINT I/O 模块添加到该 ControlLogix 组态，可以添加附加网络，然后使用 DIO 向导为该网络指定 I/O。继续阅读本文，将了解向导如何提示完成所有这些选项，以提供易于执行的工作流来创建具有本地 I/O 和分布式 I/O 的完整子系统。以下是具有 ControlLogix I/O 网络和两个附加网络的子系统示例，如图 26-17 所示。

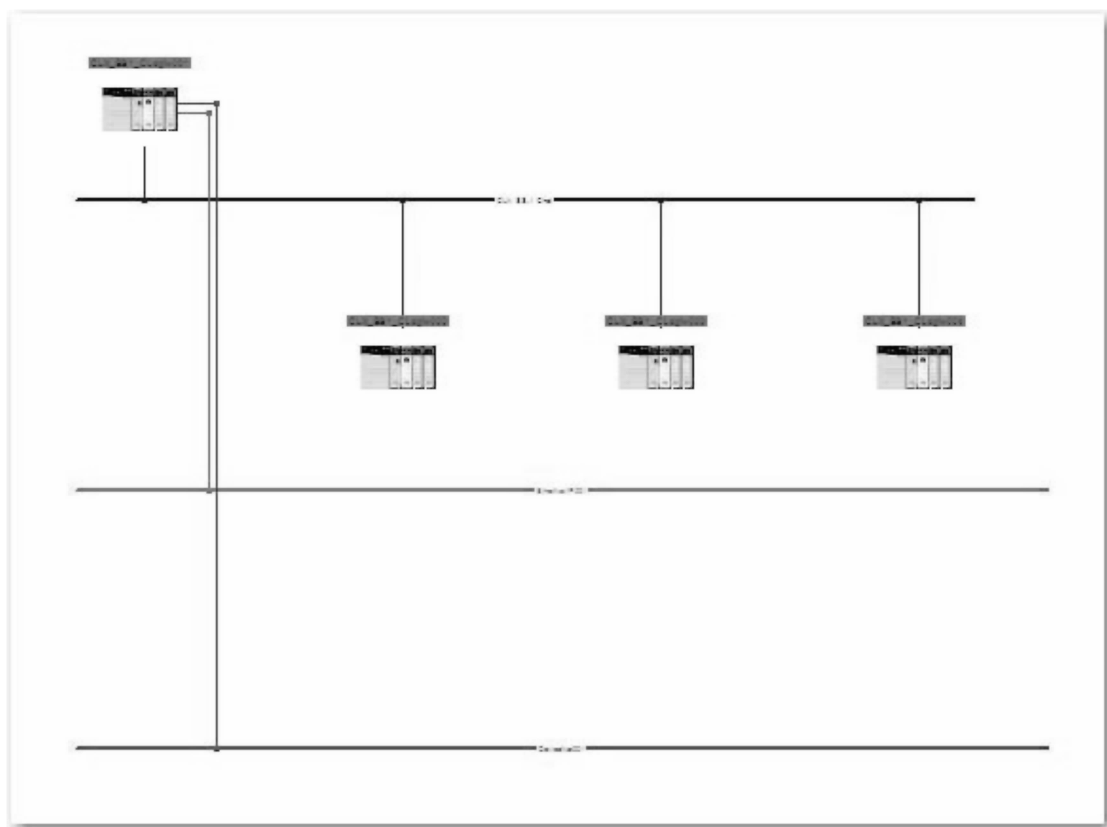


图 26-17　网络和两个附加网络的子系统示例图

6. 在“ControlLogix 机架与 I/O 网络首选项”（ControlLogix Chassis & I/O Network Preferences）页面上，大致看一下“ControlLogix 机架首选项”（ControlLogix Chassis Preferences）部分，如图 26-18 所示。

在“ControlLogix 机架首选项”（ControlLogix Chassis Preferences）部分中，可设置以下子系统总体首选项：

1）XT 宽温模块（XT Extreme Temperature）：该向导将选择 XT 模块、ControlLogix 机架、控制器和通信模块。

2）使用冗余控制器机架（Use Redundant Controller Chassis）：如果选中该框，向导将创建一个具有所需重复模块和电缆的冗余 ControlLogix 机架对。

3）允许将 I/O 置于控制器机架中（Allow I/O in Controller Chassis）：如果未选中该框，则 IAB 不会将任何 I/O 模块置于拥有控制器模块的同一机架中。相反，会将 ControlLogix I/O 置于指

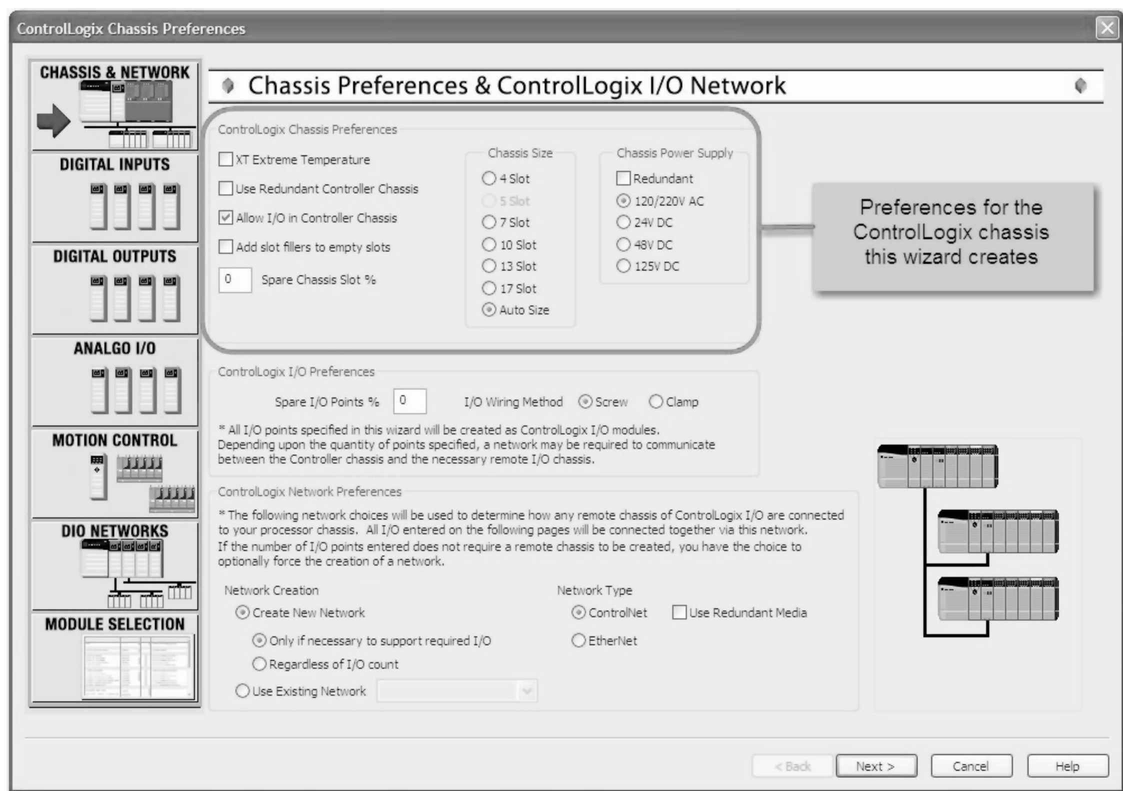


图 26-18 机架首选项

定网络上的远程机架中。如果选中该框，向导会在放置所有所需通信模块和运动控制模块之后，将尽可能多的 I/O 置于控制器机架中，然后将其他模块放置在远程机架中。

4）将槽盖板添加到空插槽（Add Slot Fillers to Empty Slots）：如果选中该框，向导会向任何不包含模块的插槽添加槽盖板（1756-N2）。

5）空闲机架插槽百分比（Spare Chassis Slot %）：指定所创建的每个机架中保留的空插槽的百分比。例如，如果将空闲插槽百分比指定为 10%，并且还指定具有 10 个插槽的机架，则 IAB 会在每个机架中保留一个空闲插槽。

6）机架大小（Chassis Size）：选择 ControlLogix 机架中的默认插槽数。自动设置功能会根据所需模块数分配机架大小。

7）机架电源（Chassis Power Supply）：选择机架电源的默认输入电压。如果选中“冗余”（Redundant）框，IAB 会将两个冗余电源以及所需电缆和连接器添加到所创建的每个机架中。

对于本实验练习，请按照图 26-19 所示设置机架首选项。

7. 仍然在“ControlLogix 机架与 I/O 网络首选项”（ControlLogix Chassis & I/O Network Preferences）页面上大致看一下“ControlLogix I/O 首选项”（ControlLogix I/O Preferences）部分，如图 26-20 所示。

1）空闲 I/O 点（Spare I/O Points）：指定 IAB 会将在该向导中指定的 ControlLogix I/O 点的百分之几自动添加到组态。例如，如果指定的空闲 I/O 点为 10%，然后在向导中指定 100 个数字量输入，则 IAB 将创建足以容纳 110 个数字量输入的模块。

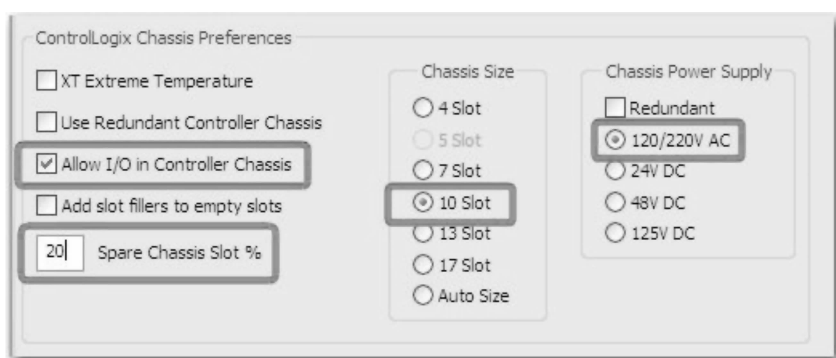


图 26-19　设置机架首选项图

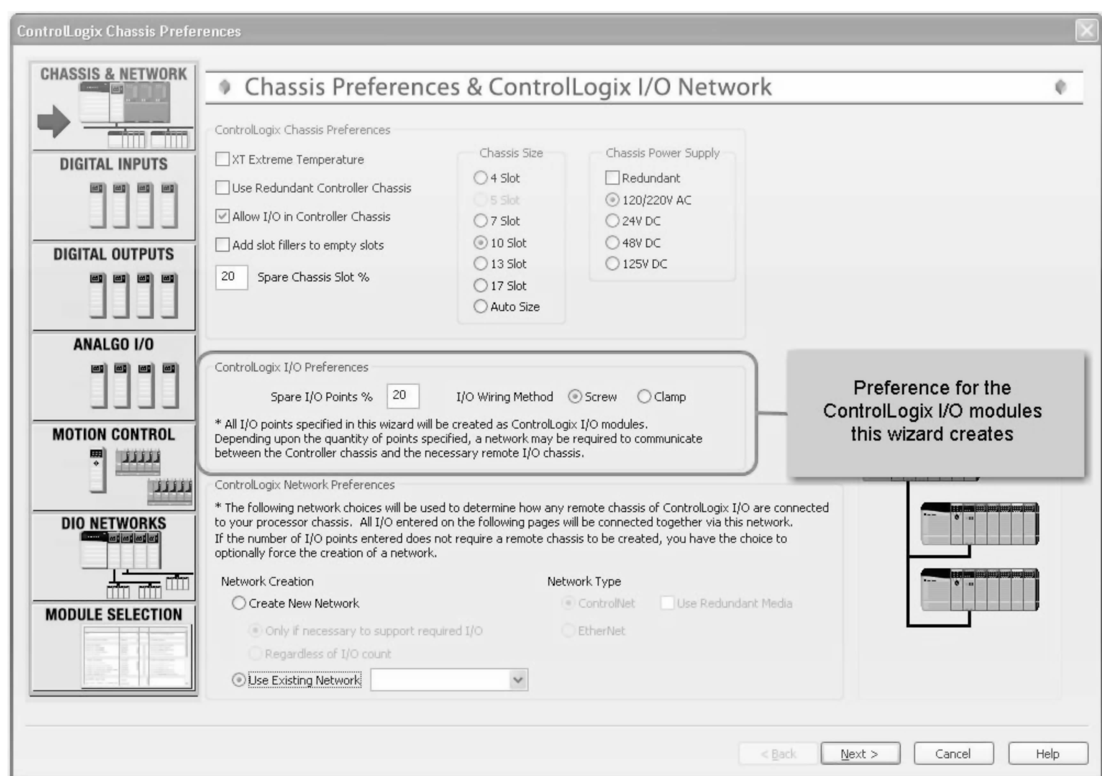


图 26-20　ControlLogix I/O Preferences 图

2）I/O 接线方法（I/O Wiring Method）：指定 IAB 将组态的接线端子的类型-螺钉型或线夹型。

在“空闲 I/O 点”（Spare I/O Points）框中输入 20%，如图 26-21 所示。

8. 仍然在“ControlLogix 机架与 I/O 网络首选项”（ControlLogix Chassis & I/O Network Preferences）页面上，大致看一下“ControlLogix 网络首选项”（ControlLogix Network Preferences）部分，如图 26-22 所示。

在这部分中，可指定：

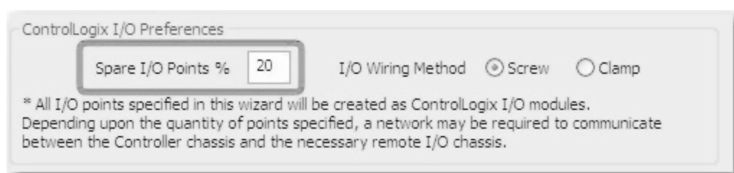


图 26-21 空闲 I/O 点设置图

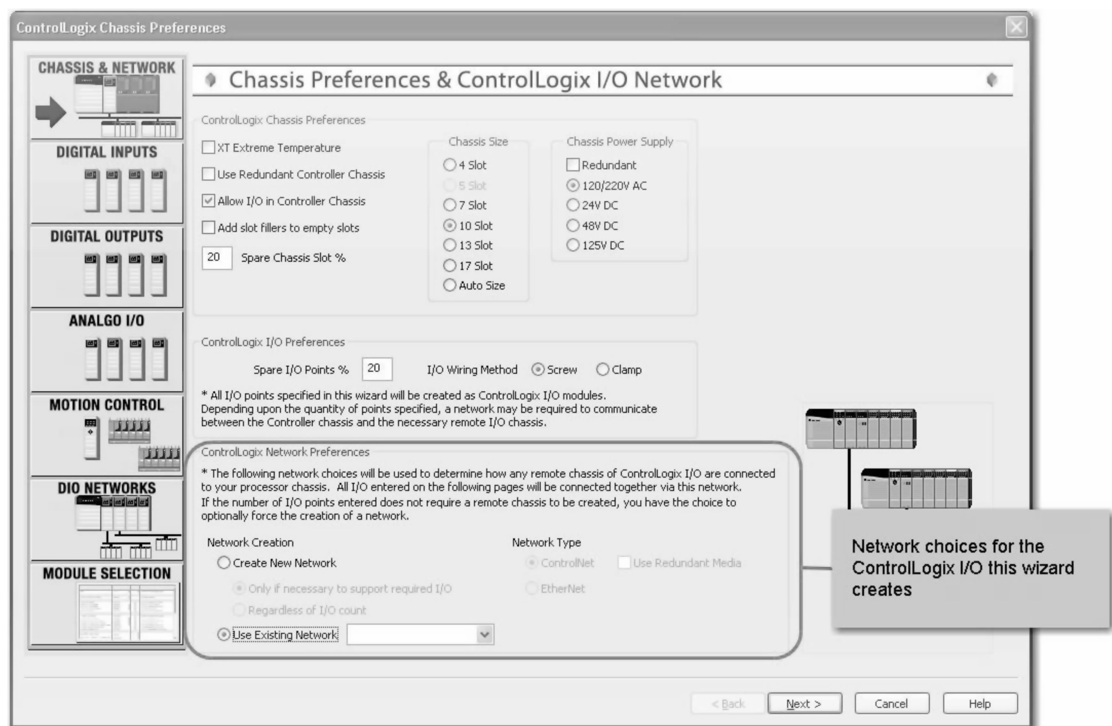


图 26-22 网络首选项

1）创建新网络还是使用 IAB 项目中已经存在的网络（由于它是新建的 IAB 项目，因此现有网络不可用）。

2）是使用 ControlNet 还是使用以太网网络。

按照图 26-23 所示设置“ControlLogix 网络首选项”（ControlLogix Network Preferences）。

单击“下一步”（Next）转到向导的下一个页面。IAB 会显示“数字量输入”（Digital Inputs）页面，如图 26-24 所示。

可以在之后出现的三个向导页面中，指定所要组态的 ControlLogix I/O 的类型和数量。可指定数字量输入、数字量输出和模拟量 I/O。

9. 在“数字量输入”（Digital Inputs）页面上，向“交流输入”（AC Inputs）中的“非隔离”（Non-Isolated）框中输入 120V。将这些输入的电压设置为 AC 24V，并选中“诊断”（Diagnostic）框。进行该设置表示，已要求向导将 1756I/O 模块组态为 12 个非隔离、24V 交流输入，并包含诊断功能，如图 26-25 所示。

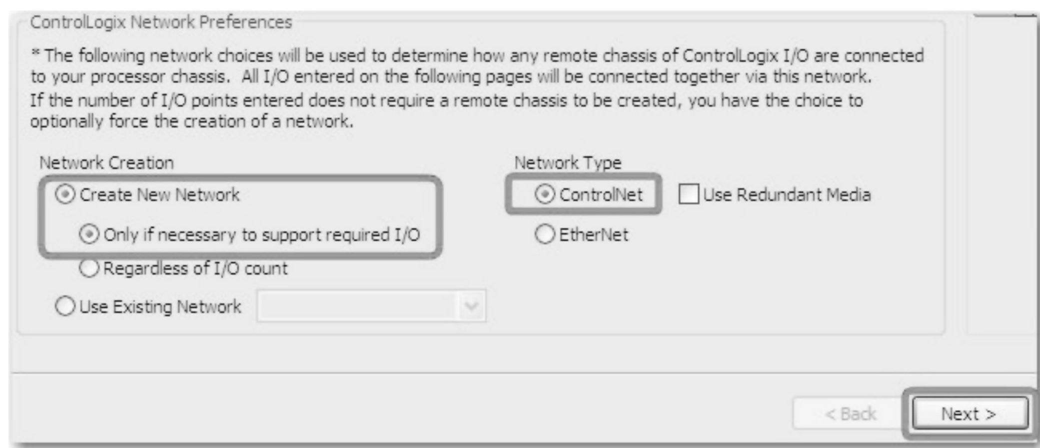


图 26-23 网络首选项设置图

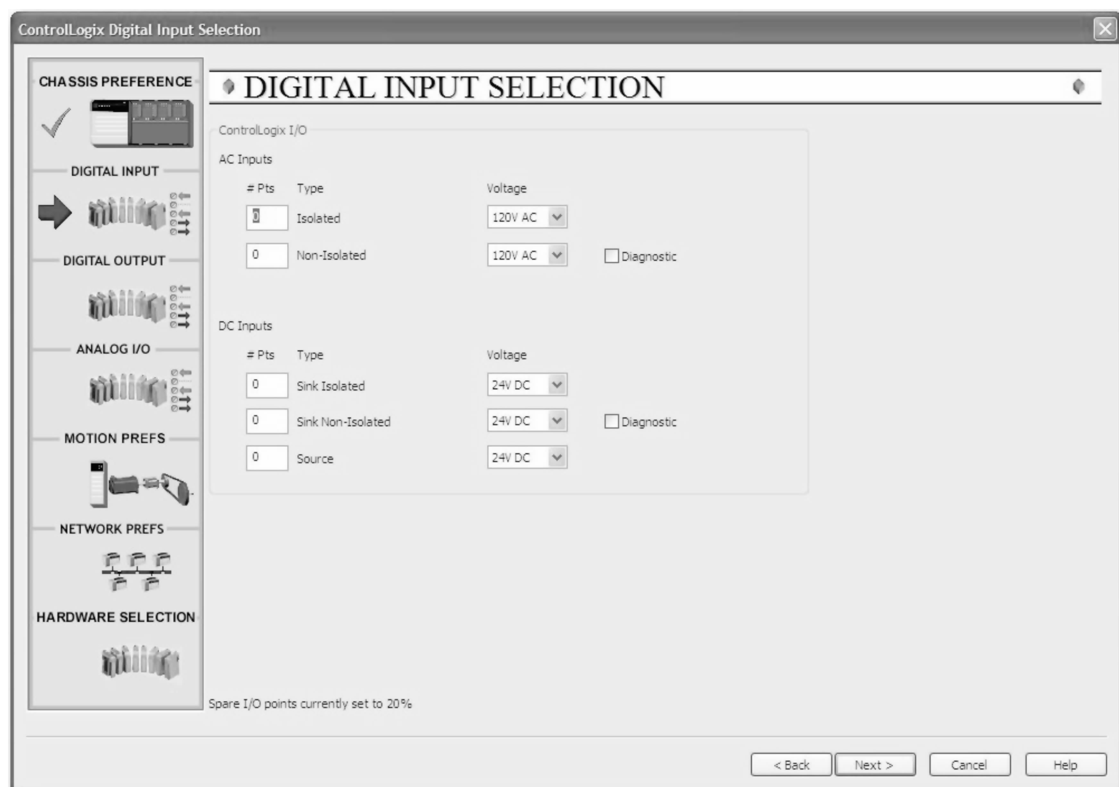


图 26-24 数字量输入图

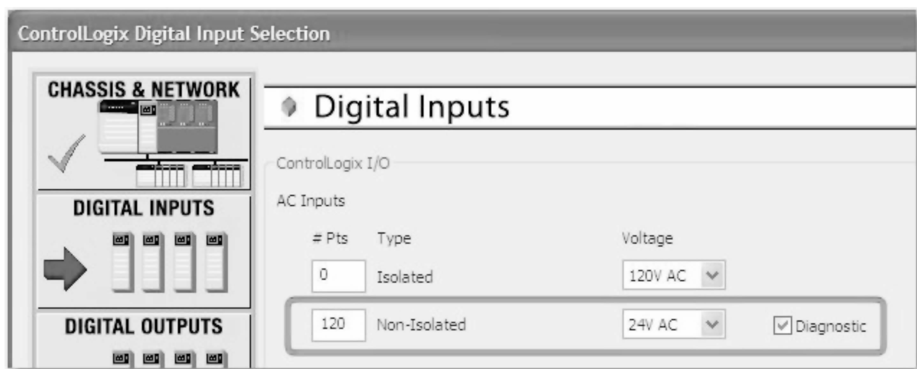


图 26-25 数字量输入设置图

单击“下一步”（Next）继续。IAB 会显示如图 26-26 所示消息。

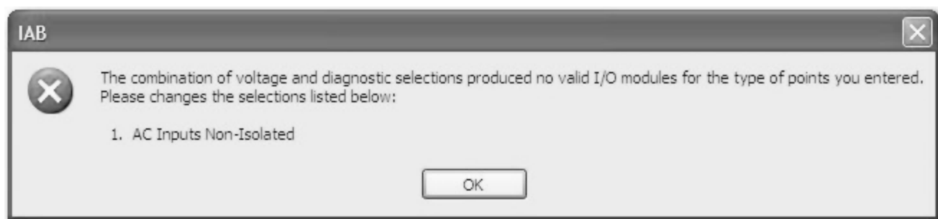


图 26-26 消息图

不存在能够提供 24V 交流非隔离输入，并具有诊断功能的 1756 ControlLogix I/O 模块。单击“确定”（OK）继续。IAB 会保留在当前页面上，以便“能够修订”的选择。将电压选项更改为 AC 120V，然后取消选中“诊断”（Diagnostic）框，如图 26-27 所示。

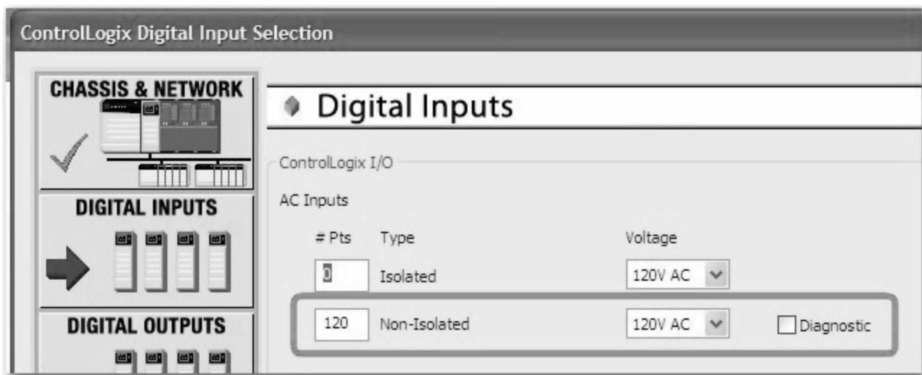


图 26-27 更改后设置图

单击“下一步”（Next）继续。IAB 会显示“数字量输出”（Digital Outputs）页面，如图 26-28 所示。

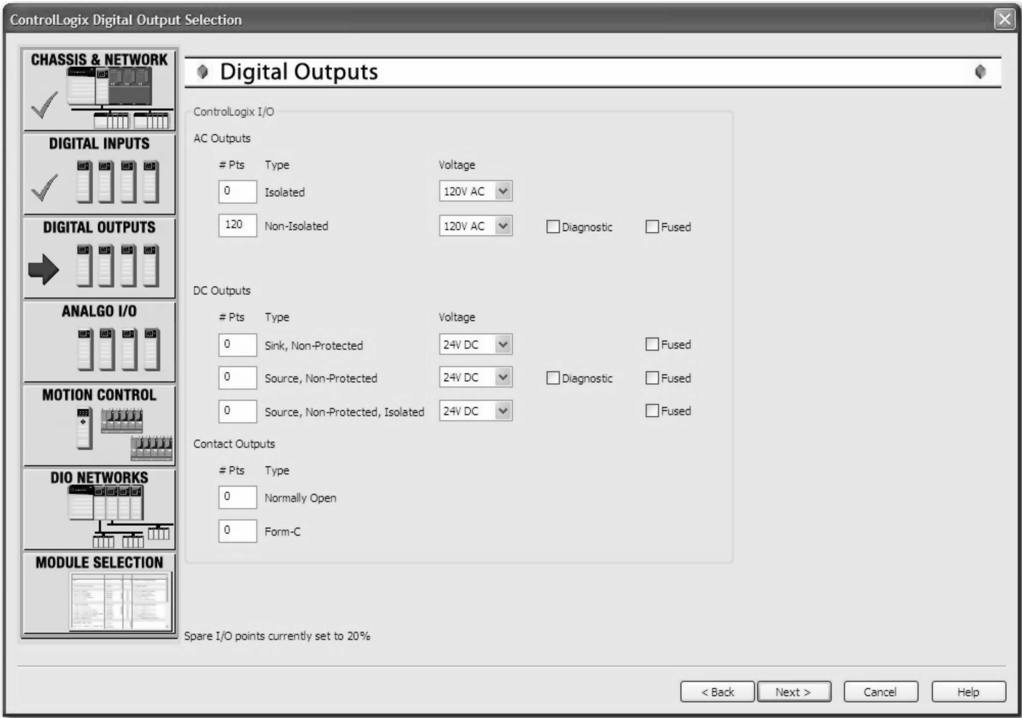


图 26-28　数字量输出图

10. 在“数字量输出”（Digital Outputs）页面上，于“交流输出”（AC Outputs）中的“非隔离”（Non-Isolated）框内输入 120V，如图 26-29 所示。

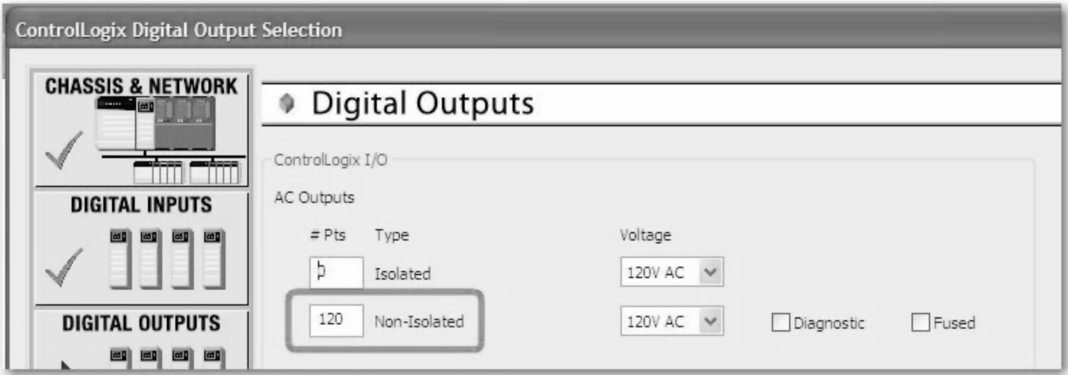


图 26-29　数字量输出设置图

单击“下一步”（Next）继续。IAB 会显示“模拟量 I/O”（Analog I/O）页面，如图 26-30 所示。

11. 在“模拟量 I/O”（Analog I/O）页面中，于“单端输入”（Single Ended inputs）框内输入 60，并于“隔离电压输出”（Isolated Voltage Outputs）框内输入 60，如图 26-31 所示。

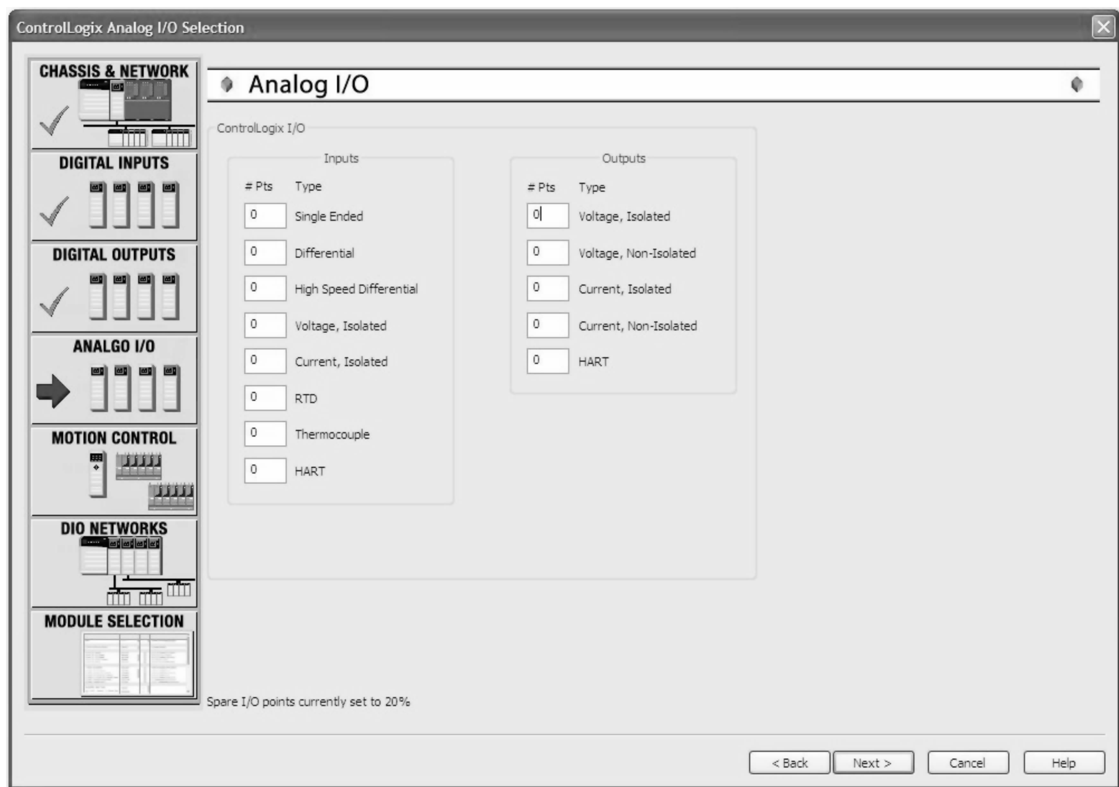


图 26-30 模拟量 I/O 图

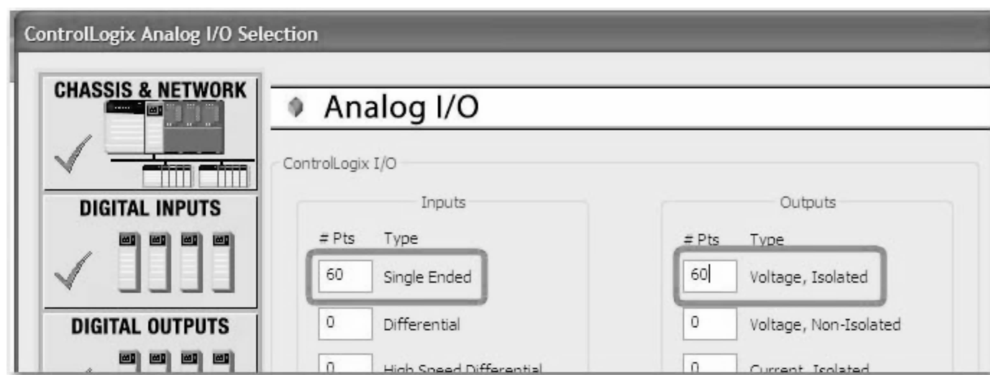


图 26-31 模拟量 I/O 设置图

单击“下一步”（Next）继续。IAB 会显示“运动控制”（Motion Control）页面，如图 26-32 所示。

12. 在“运动控制”（Motion）页面中，向“轴数”（Number of Axes）框内输入 12。对于“应用等级”（Level of Application），请选择“中等”（Medium）单选按钮（见图 26-33）。之后，将能够选择在向导中选取哪一个运动控制模块。

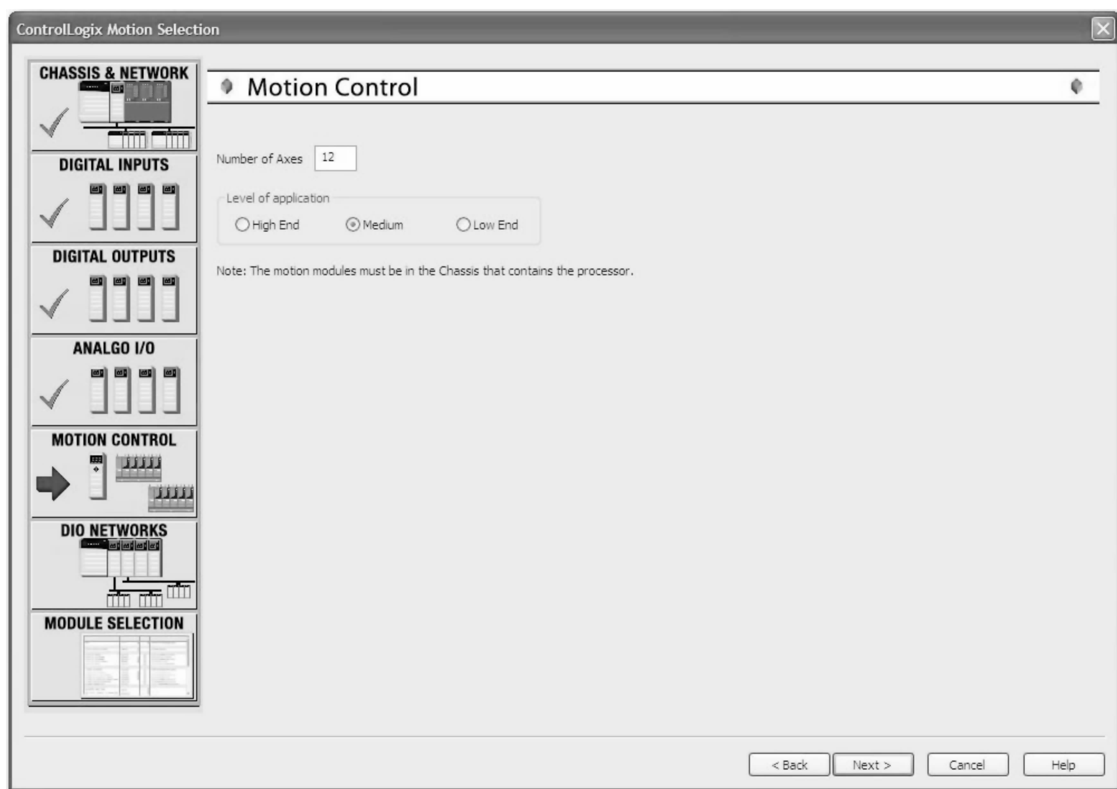


图 26-32　运动控制图

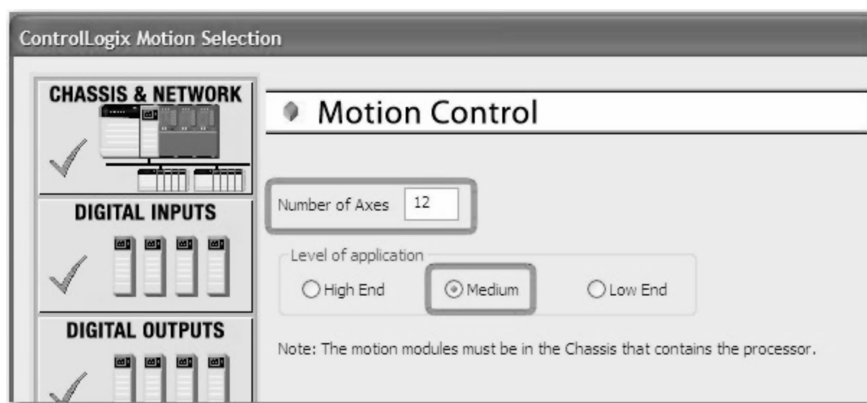


图 26-33　运动控制设置图

“应用等级”（Level of Application）设置会影响 IAB 为运动控制进行的控制器加载计算。这是一种经验法则计算。

单击“下一步”（Next）继续。IAB 会显示“分布式 I/O 与附加网络模块的网络”（Networks for Distributed I/O & Additional Network Modules）页面，如图 26-34 所示。

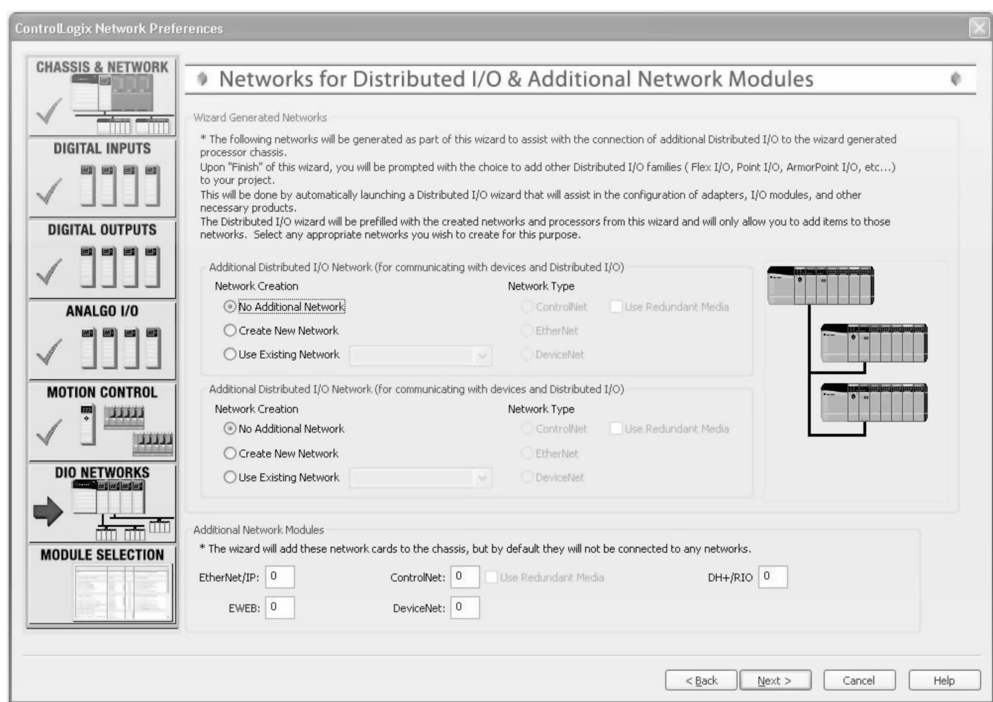


图 26-34　分布式 I/O 与附加网络模块的网络图

13. 查看该向导页面并阅读该页面上的文本注释。

在该页面上：

1）除了之前指定的 ControlLogix I/O 网络以外，还可为分布式 I/O 指定多达两个网络。这些附加网络可以是新网络，也可以是 IAB 项目中已经存在的网络。IAB 将为 指定的网络添加网络模块，并将其连接到指定的网络，如图 26-35 所示。

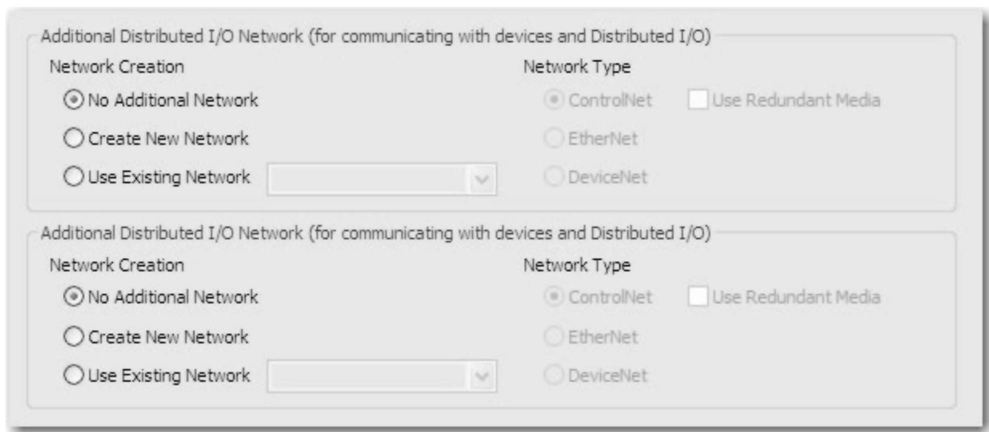


图 26-35　附加网络图

2）可以指定要由该向导创建的更多的附加通信模块。这些模块将被置于 ControlLogix 机架中，但不会连接到任何网络，如图 26-36 所示。

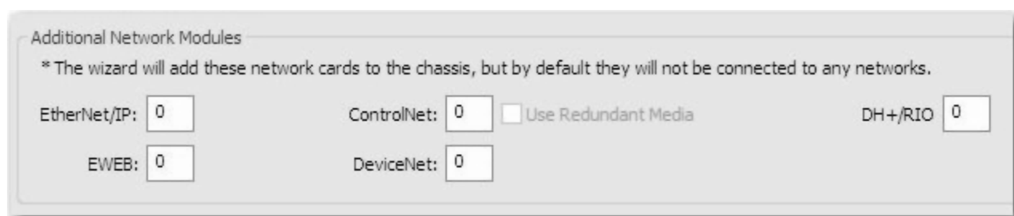


图 26-36　附加通信模块图

14. 按照以下方法添加两个附加的分布式 I/O 网络：

1）创建新网络 EtherNet/IP；

2）创建新网络 DeviceNet，如图 26-37 所示。

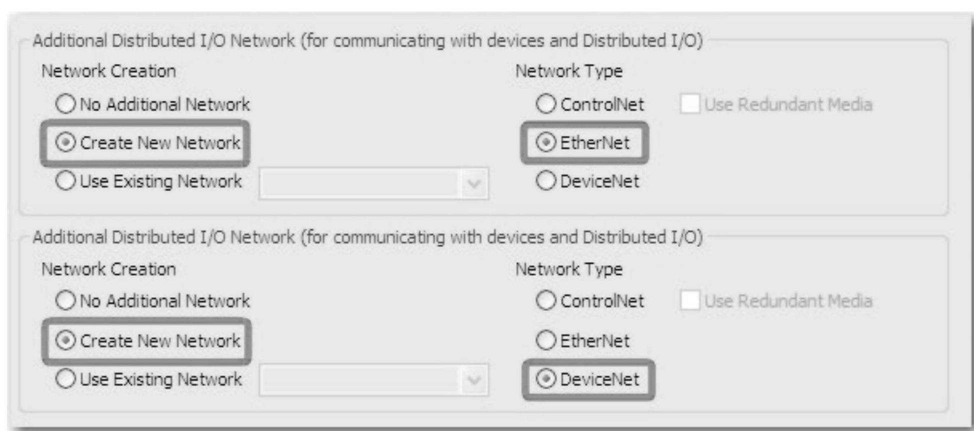


图 26-37　添加两个附加网络图

15. 添加一个附加 EWEB 模块。单击“下一步”（Next）继续，如图 26-38 所示。

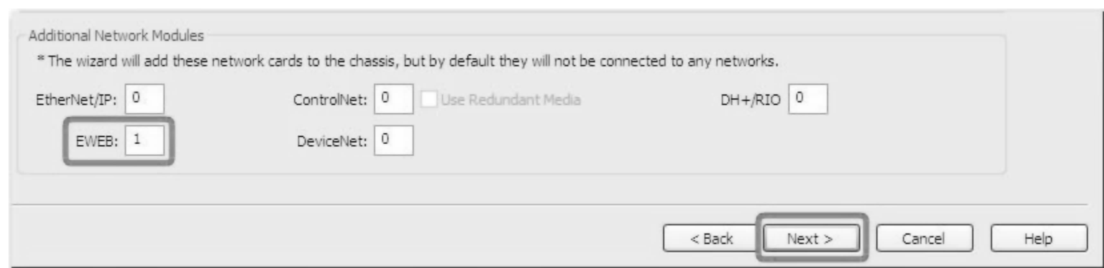


图 26-38　附加 EWEB 模块

IAB 会显示“硬件选项”（Hardware Selection）页面，如图 26-39 所示。

16. 在页面顶部，IAB 会显示正在组态的 I/O 平台（本示例中为 ControlLogix I/O）的一些选项。该信息会影响 IAB 评估组态性能的方式，如图 26-40 所示。

1）额外内存和任务数（Extra memory and number of tasks）：如果知道控制器将被设计为执行多少个任务，或知道除了用于应用特定数据的程序所需的内存之外还需要使用多少内存，则可在

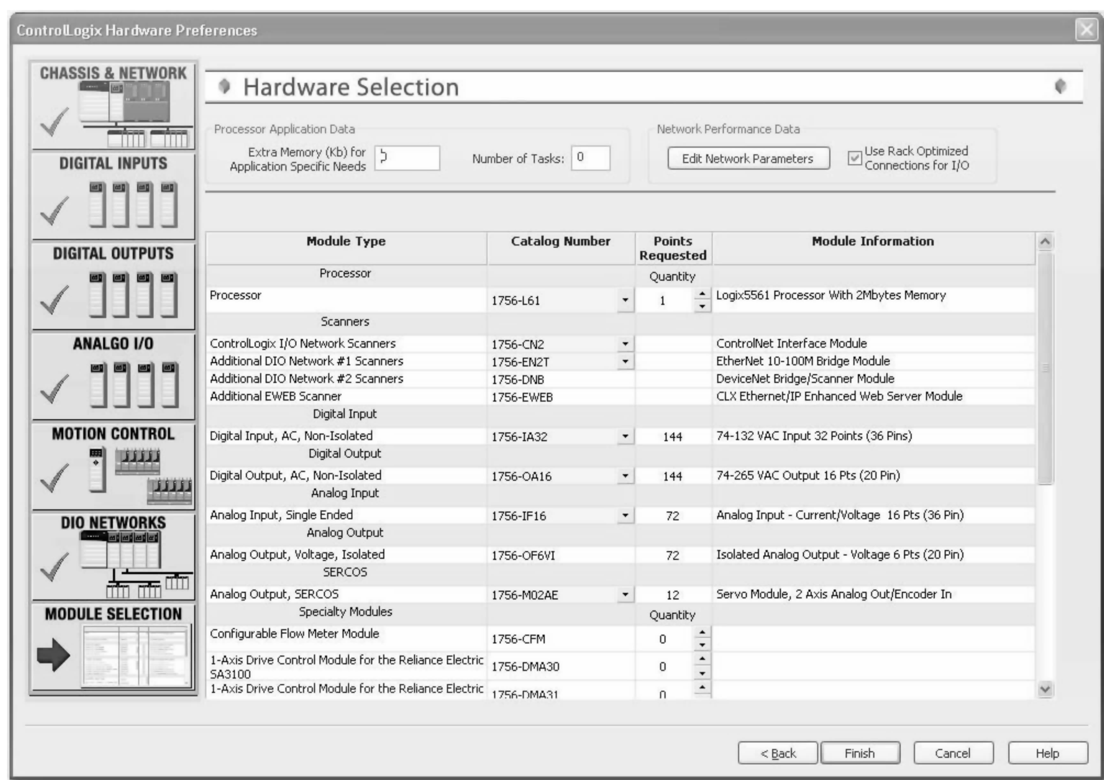


图 26-39　硬件选项图

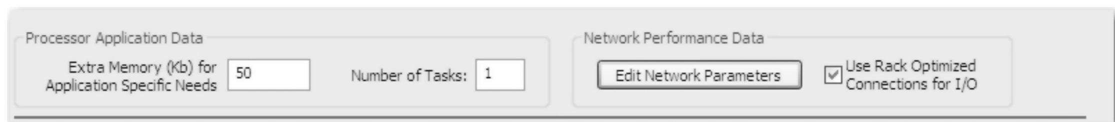


图 26-40　组态的 I/O 平台图

此输入该值。否则，仅可使用默认值。对于本实验，请使用默认值。

2）为 I/O 使用机架优化连接（Use Rack Optimized Connections for I/O）：如果选中该框（默认选项），IAB 会为每个机架的数字量 I/O 分配一条连接，还会为每个模拟量 I/O 模块分配一条连接，并会为专用 I/O 模块分配其所需的任何连接。这样会减少所需连接的总数。如果取消选中该框，IAB 会为远程 I/O 机架中的每个模块（包括数字量 I/O 模块）分配一条连接。

3）编辑网络参数（Edit Network Parameters）：单击“编辑网络参数”（Edit Network Parameters）按钮。IAB 会显示“高级首选项”（Advanced Performance）对话框。在该对话框中，可以指定要求使用快速、中速和慢速更新时间的总数字量 I/O 和模拟量 I/O 的百分比，还可以指定每种类型的特定 RPI（请求信息包间隔）。另外，也可指定网络的生产者标签的数目和大小以及一些用于其他需求的预留控制器连接。IAB 可以使用这些数字来计算网络性能。对于该基本实验，不会更改任何默认设置。仅需了解这些设置已存在并且当 IAB 检验性能时，这些设置会被使用即可。单击“确定”（OK）关闭对话框，如图 26-41 所示。

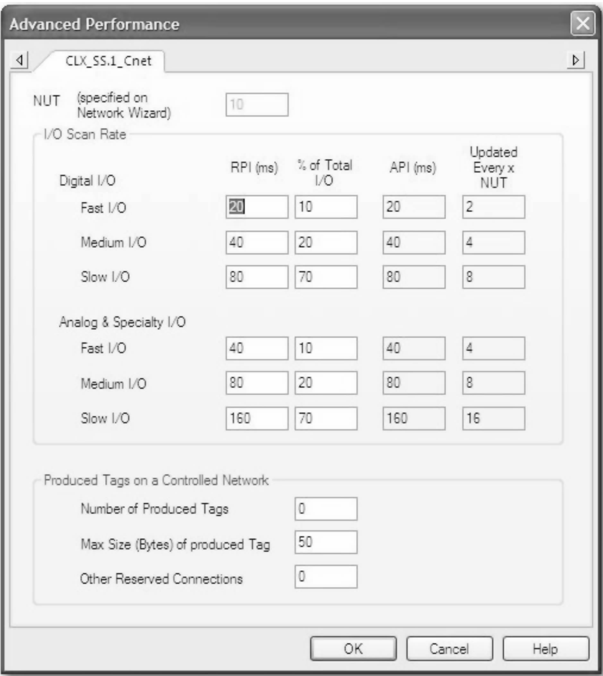


图 26-41　设置对话框

17. 向导模块选型（Wizard module selections）。在该区域中，会列出在向导中指定的所有要求，以及向导所选择的、用于满足这些要求的模块，如图 26-42 所示。

Module Type	Catalog Number	Points Requested	Module Information
Processor		Quantity	
Processor	1756-L61	1	Logix5561 Processor With 2Mbytes Memory
Scanners			
ControlLogix I/O Network Scanners	1756-CN2		ControlNet Interface Module
Additional DIO Network #1 Scanners	1756-EN2T		EtherNet 10-100M Bridge Module
Additional DIO Network #2 Scanners	1756-DNB		DeviceNet Bridge/Scanner Module
Additional EWEB Scanner	1756-EWEB		CLX Ethernet/IP Enhanced Web Server Module
Digital Input			
Digital Input, AC, Non-Isolated	1756-IA32	144	74-132 VAC Input 32 Points (36 Pins)
Digital Output			
Digital Output, AC, Non-Isolated	1756-OA16	144	74-265 VAC Output 16 Pts (20 Pin)
Analog Input			
Analog Input, Single Ended	1756-IF16	72	Analog Input - Current/Voltage 16 Pts (36 Pin)
Analog Output			
Analog Output, Voltage, Isolated	1756-OF6VI	72	Isolated Analog Output - Voltage 6 Pts (20 Pin)
SERCOS			
Analog Output, SERCOS	1756-M02AE	12	Servo Module, 2 Axis Analog Out/Encoder In

图 26-42　向导模块选型图

1）更改向导选择的处理器：可以更改所选的处理器以及处理器的数目，如图 26-43 所示。

不要在此进行任何更改；仅使用向导选择的值即可。

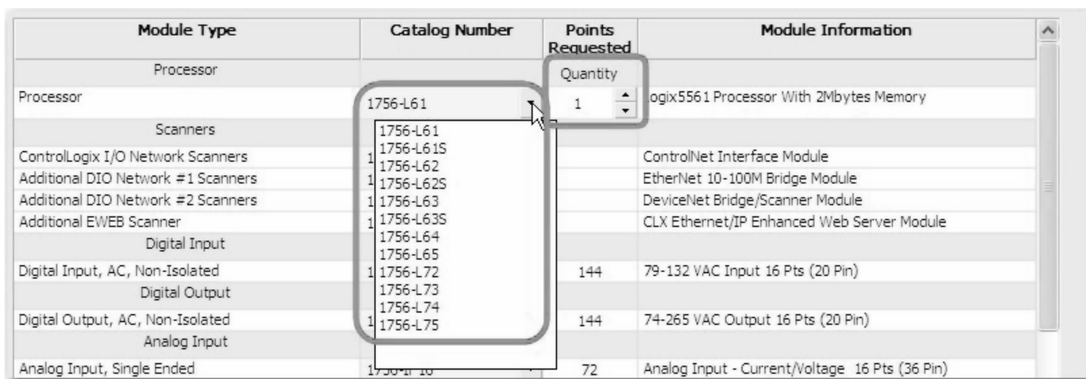


图 26-43 更改向导选择的处理器

2）更改向导选择的模块：可在“选择的模块”（Selected Modules）区域的“目录号”（Catalog Number）列中，更改 IAB 选项以符合的要求。在标为“数字量输入，交流，非隔离”（Digital Input，AC，Non-Isolated）的行中，请注意，IAB 已选择了 1756-IA32 模块。单击模块目录号旁的下拉列表，从列表中选择 1756-IA16。这会显示如何选择 IAB 所选模块的替代模块，如图 26-44 所示。

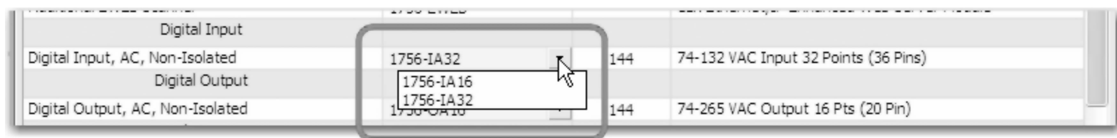


图 26-44 更换数字量模块

在“模拟量输出，SERCOS”（Analog Output，SERCOS）一行中，单击下拉箭头，并为所需运动的 12 个轴选择 1756-M16SE 模块，如图 26-45 所示。

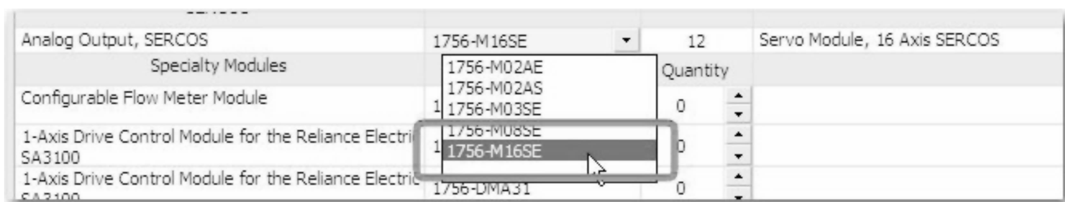


图 26-45 轴选择图

IAB 不会自动选择运动控制模块。始终应记住选中该选项，并根据控制对象的要求选择相应的模块。

18. 专用模块（Specialty Modules）。在向导选择的模块列表下方，会列出 1756 系列中的所有专用模块（见图 26-46）。可以指定每种要包含在组态中的模块数量。

向下滚动到“专用模块”（Specialty Module）区域，然后通过单击向上微调箭头的方式选择一个“1756-HSC 高速计数器”（1756-HSC High Speed Counter）模块，如图 26-47 所示。

19. 单击“完成”（Finish）按钮，完成向导。处理完所有选项之后，IAB 会显示如图 26-48 所示消息。

Specialty Modules		Requested Quantity
Configurable Flow Meter Module	1756-CFM	0
1-Axis Drive Control Module for the Reliance Electric SA3100	1756-DMA30	0
1-Axis Drive Control Module for the Reliance Electric SA3100	1756-DMA31	0
1-Axis Drive Control Module for the Reliance Electric SD3000	1756-DMD30	0
1-Axis Regen Field Control Module for the Reliance Electric SF3000	1756-DMF30	0
FactoryTalk Historian Machine Edition- 1GB	1756-HIST1G	0
FactoryTalk Historian Machine Edition- 2GB	1756-HIST2G	0
2 Channel / 4 Output High Speed Counter (36 Pin)	1756-HSC	0
2 Channel LDT Hydraulics Module (36 Pin)	1756-HYD02	0
Multi-Vendor Interface Module	1756-MVI	0
Empty Slot Filler for 1756 Chassis	1756-N2	0
Empty Slot Filler for 1756 XT Chassis	1756-N2XT	0
Programmable Limit Switch Module	1756-PLS	0
SynchLink Module	1756-SYNCH	0
Time synchronization using Global Positioning System (GPS).	1756HP-TIME	0

图 26-46 专用模块

FactoryTalk Historian Machine Edition- 1GB	1756-HIST1G	0
FactoryTalk Historian Machine Edition- 2GB	1756-HIST2G	0
2 Channel / 4 Output High Speed Counter (36 Pin)	1756-HSC	1
2 Channel LDT Hydraulics Module (36 Pin)	1756-HYD02	0
Multi-Vendor Interface Module	1756-MVI	0

图 26-47 高速计数器

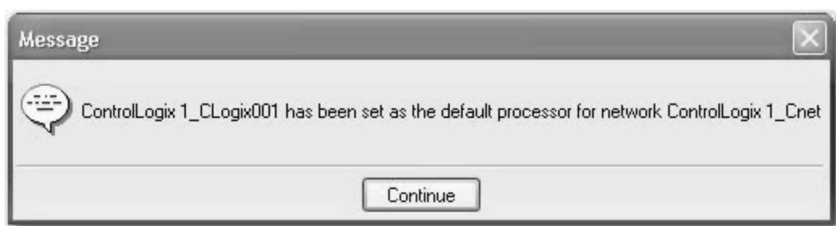


图 26-48 消息图

该消息通知系统会自动将该向导创建的处理器分配为在向导所创建的网络上创建的所有 I/O 的控制器。该操作会为 提供方便，并且对检查组态性能十分重要。稍后，将了解到如何更改处理器和 I/O 分配。

单击“继续”（Continue）。IAB 会显示如图 26-49 所示对话框。

可以通过该对话框在 ControlLogix 向导中指定的网络上添加分布式 I/O。如果选择“是”（Yes），IAB 将启动分布式 I/O 向导，并且将指定通过 ControlLogix 向导创建的三个网络之一的分布式 I/O 要求。IAB 将创建在网络上指定的分布式 I/O，并会自动将 I/O 分配给 ControlLogix 向导

创建的 ControlLogix 处理器。

20. 单击“是”（Yes）。IAB 会立即启动 DIO 向导并显示“网络与系列首选项”（NETWORK & FAMILY PREFERENCES）页面，（如图 26-50 所示）。

21. 请注意，“网络类型”（Network Type）将灰显，且“网络生成”（Network Generation）区域中的“创建新网络”（Create New Network）选项也会灰显。由于，是从 ControlLogix 向导中启动的 DIO 向导的这一实例，因此仅可选择在 ControlLogix 向导中指定的网络。在“使用现有网络”（Use Existing Network）下拉列表中，请注意由 ControlLogix 向导创建的三个网络所列出的方式：

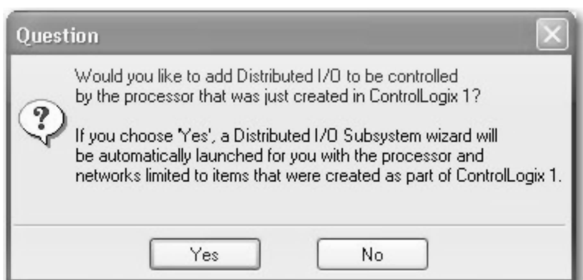


图 26-49 启动分布式 I/O 向导

1）ControlLogix 1_Cnet：用于 ControlLogix I/O 的 ControlNet 网络（见图 26-51）。

2）DeviceNet001 和 EtherNet/IP001：指定的附加 DIO 网络，如图 26-51 所示。

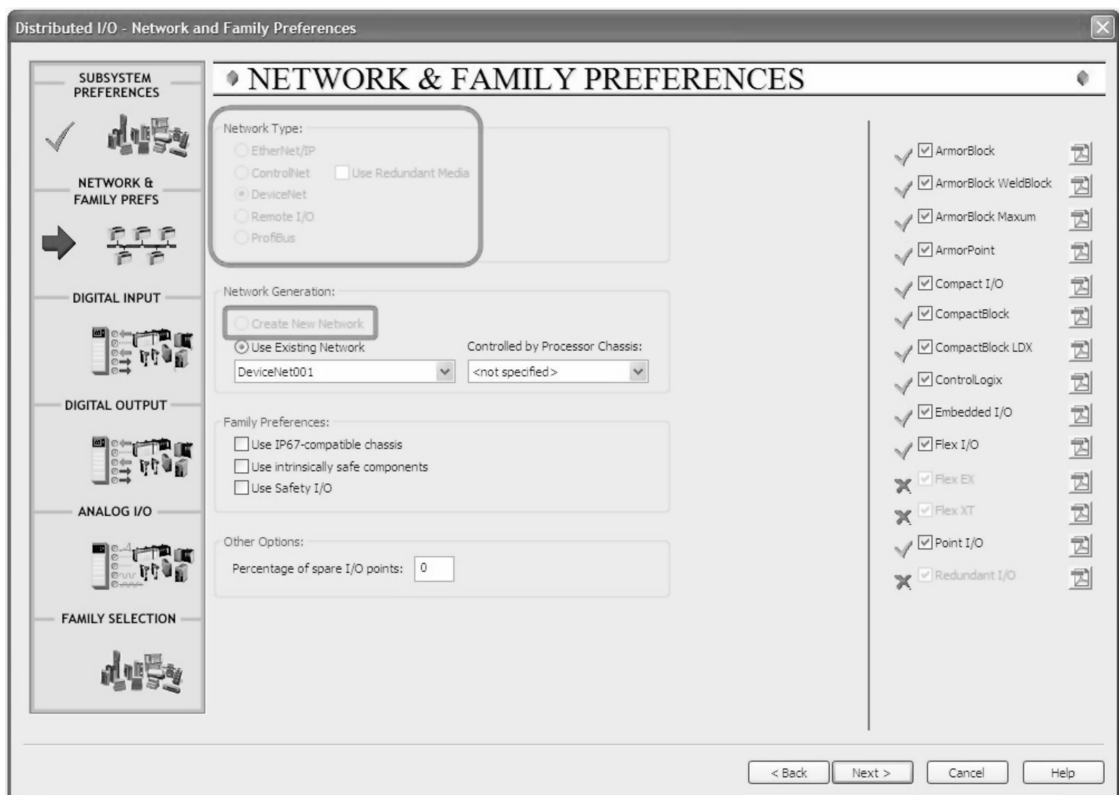


图 26-50 网络与系列首选项

选择 EtherNet/IP001。在“由控制器机架控制”（Controlled by Processor Chassis）下拉列表中：请注意，仅会列出由 ControlLogix 向导创建的控制器。单击“下一步”（Next）继续。IAB 会显示“数字量输入选择”（DIGITAL INPUT SELECTION）页面，如图 26-52 所示。

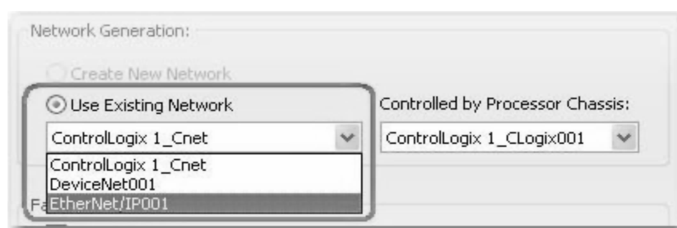


图 26-51　生成网络

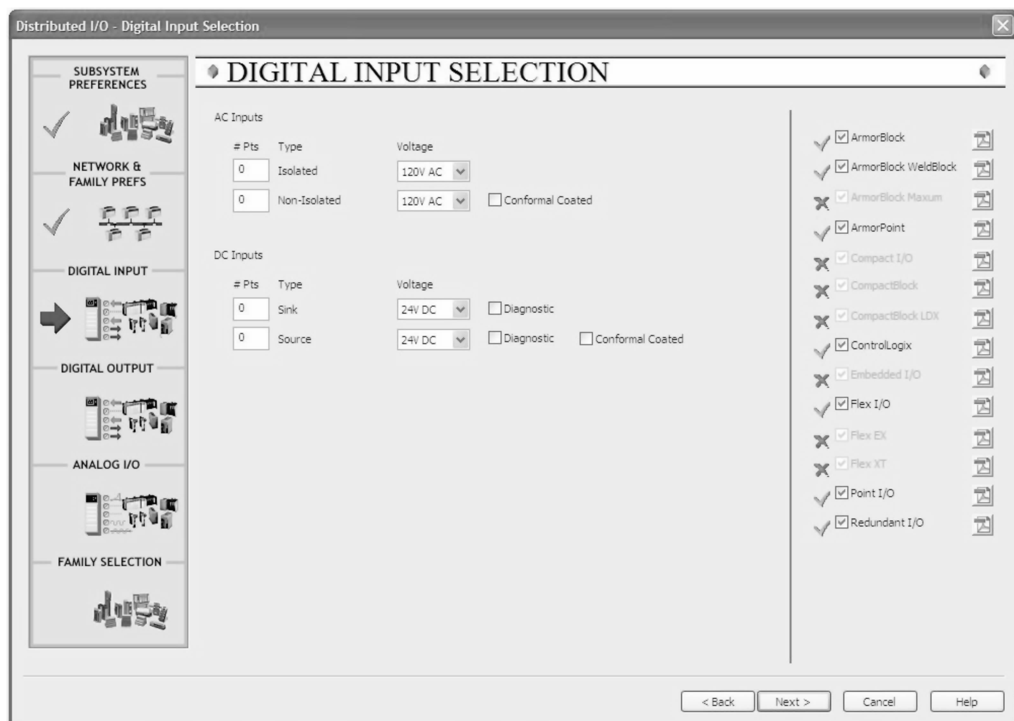


图 26-52　数字量输入选择

22. 请注意，向导右侧的 DIO 平台列表已改为只显示那些与所选网络 EtherNet/IP 兼容的平台。当在向导中进行选择时，平台选择将改为显示与选择兼容的平台，以及不兼容的平台。单击其中一个 pdf 图标，以显示对应的 DIO 平台的文档。先关闭该文档，然后再继续进行，如图 26-53 所示。

按照图 26-54 所示输入数字量输入。

单击“下一步”（Next）继续。IAB 会显示“数字量输出选择”（DIGITAL OUTPUT SELECTION）页面，如图 26-55 所示。

23. 按照图 26-56 所示选择数字量输出。

如果选中“涂层防护”（Conformal Coated）框，则 DIO 平台均不符合要求。请取消选中“涂层防护”（Conformal Coated）框并单击“下一步”（Next）以继续。IAB 会显示“模拟量 I/O 选择”（ANALOG I/O SELECTION）对话框，如图 26-57 所示。

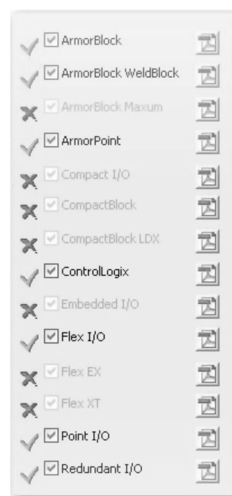


图 26-53　pdf 图标

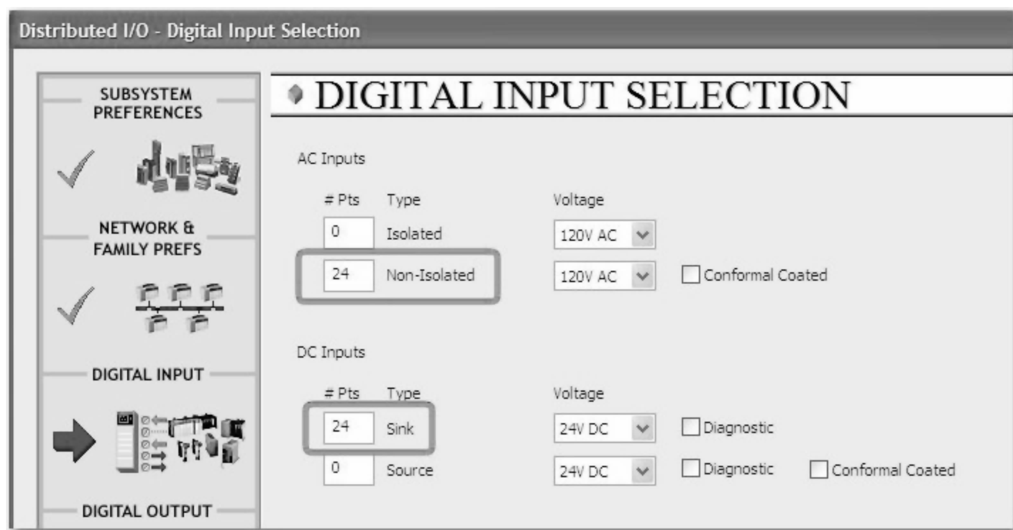


图 26-54 输入数字量输入

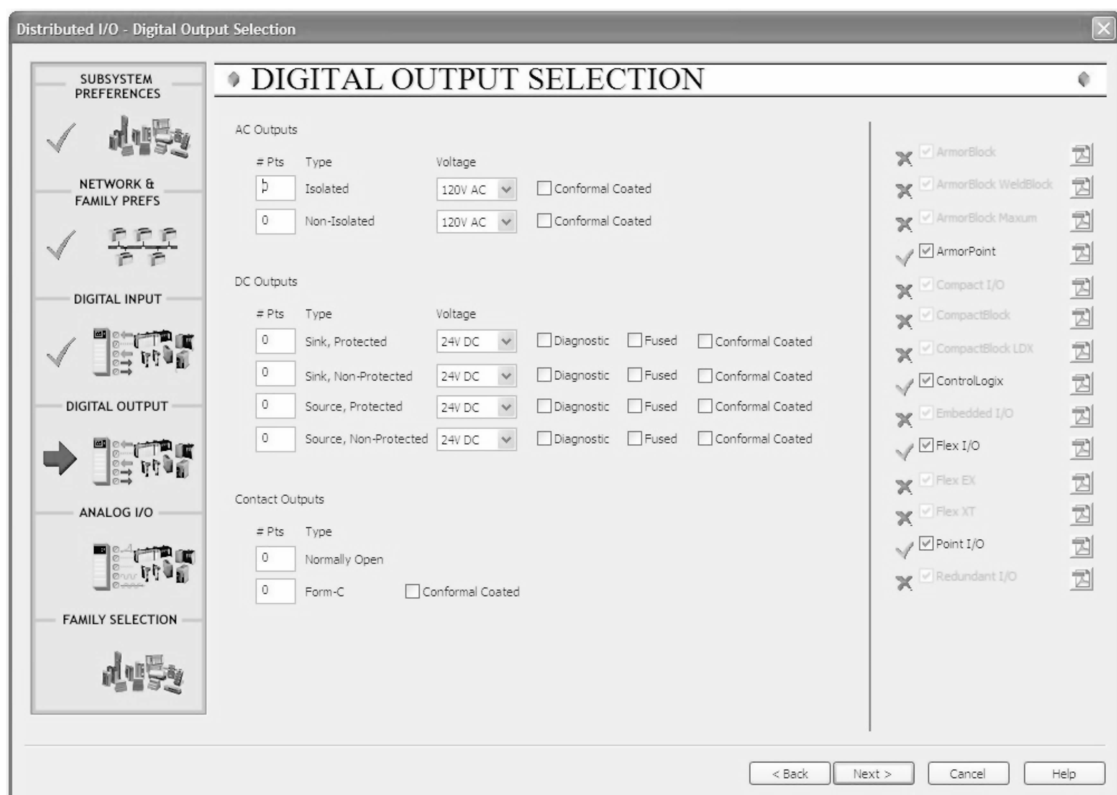


图 26-55 数字量输出选择

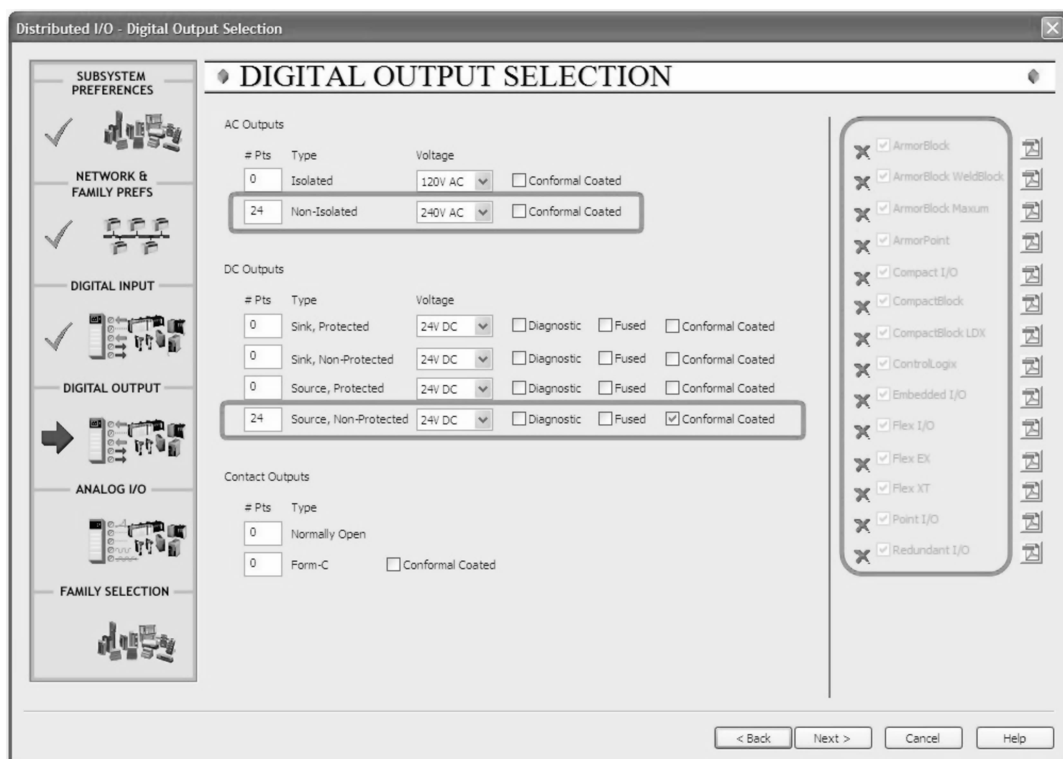


图 26-56　数字量输出选择

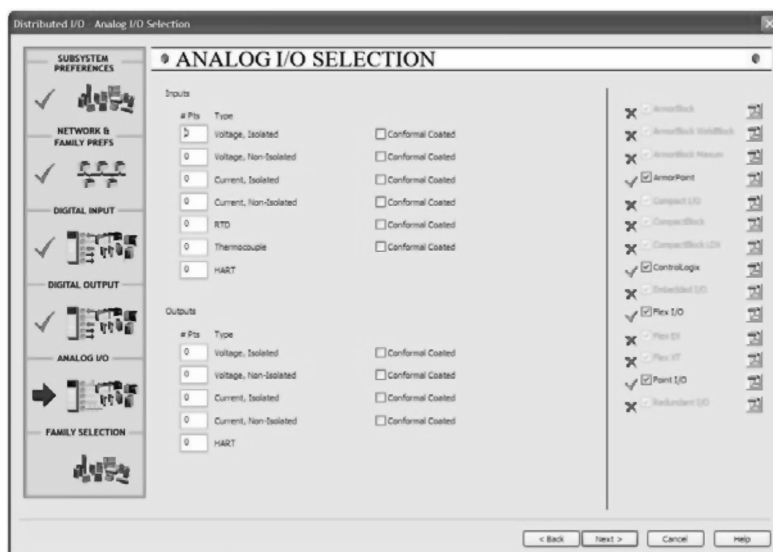


图 26-57　模拟量 I/O 选择

24. 在“电流，非隔离输入”（Current, Non-Isolated Inputs）中输入 25，在“电流，非隔离输出”（Current, Non-Isolated output）中输入 25，如图 26-58 所示。

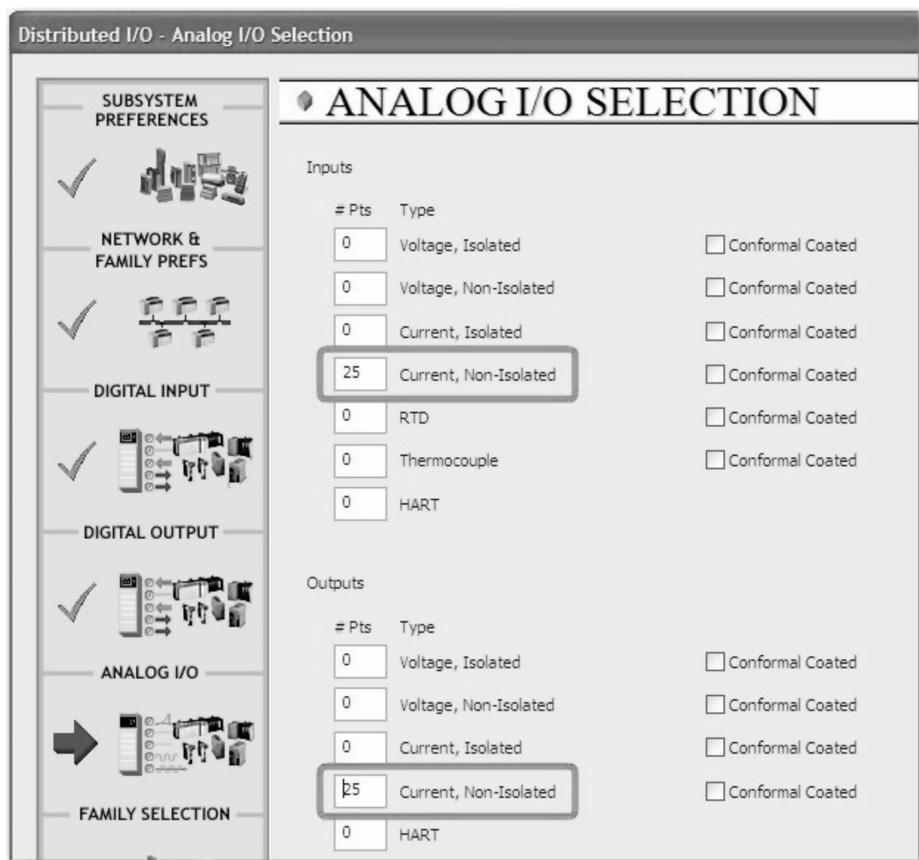


图 26-58 模拟量设定

单击“下一步”（Next）继续。IAB 会显示“系列选择/创建选项”（FAMILY SELECTION/BUILD UP OPTIONS）页面，如图 26-59 所示。

该页面的外观与 ControlLogix 向导中的“硬件首选项”（Hardware Preferences）页面相似，基本上具有相同的组织结构和功能。它会在顶部列出平台特定的选项，在中间部分列出模块选择，在底部列出专用模块。

25. 取消选中“使用与处理器之间的机架优化连接”（Use rack optimized connection to the processor）选项，如图 26-60 所示。

26. 单击向导上方的“系列”（Family）下拉箭头，如图 26-61 所示。

该下拉列表会列出单击“下一步”（Next）转到此界面之前所选中的所有 DIO 平台。单击“后退”（Back）按钮返回“模拟量 I/O 选择”（Analog I/O Selection）界面。请注意，应选中“系列”（Family）下拉列表中显示的四个平台。取消选中 ControlLogix I/O 框，如图 26-62 所示。

单击“下一步”（Next）返回“系列选择/创建选项”（FAMILY SELECTION/BUILD UP OPTIONS）界面。

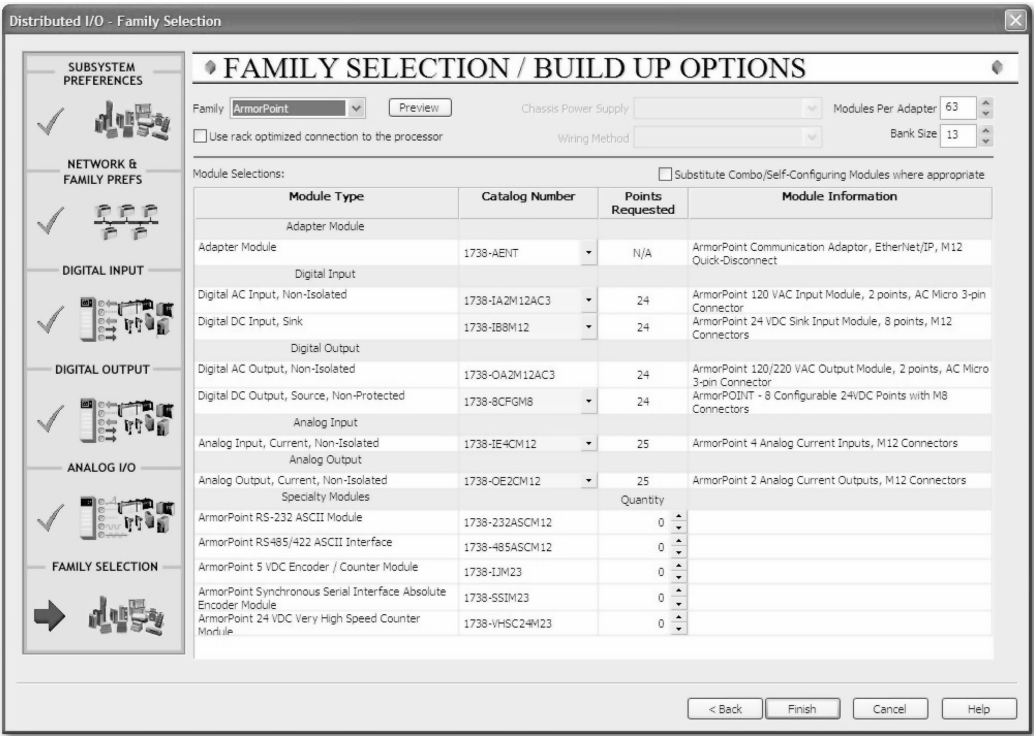


图 26-59 系列选择/创建选项

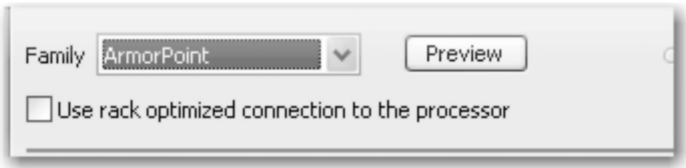


图 26-60 使用与处理器之间的机架优化连接



图 26-61 系列选择

27. 再次单击“系列”（Family）下拉列表，请注意，不会再显示 ControlLogix I/O 平台，如图 26-63 所示。

使用 DIO 向导时，请记得务必选中所感兴趣的平台。

目前，选择了 ArmorPoint 平台。该表会列出所选的 ArmorPoint 模块。单击系列名称旁的“预览”（Preview）按钮，如图 26-64 所示。



图 26-62 取消选中 ControlLogix I/O 框

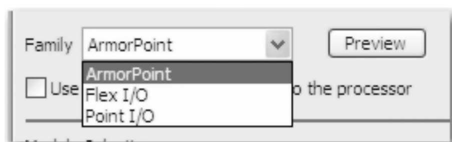


图 26-63 系列选择图

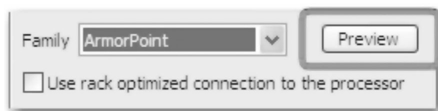


图 26-64 预览选项

IAB 会显示由向导指定的 ArmorPoint I/O 的物料清单（包括标价），如图 26-65 所示。

Quick Bill Of Material

Qty	Catalog #	Description	Unit Price	Price
003	1738-8CFGM8	ArmorPOINT - 8 Configurable 24VDC Points with M8 Connectors	208.00	624.00
002	1738-EXT1	ArmorPoint Bus Extension Base, 1 Meter	143.00	286.00
007	1738-IE4CM12	ArmorPoint 4 Analog Current Inputs, M12 Connectors	510.00	3570.00
013	1738-OE2CM12	ArmorPoint 2 Analog Current Outputs, M12 Connectors	305.00	3965.00
002	1738-EP24DC	ArmorPoint 24 VDC Expansion Power Unit	180.00	360.00
012	1738-OA2M12AC3	ArmorPoint 120/220 VAC Output Module, 2 points, AC Micro 3-pin Connector	180.00	2160.00
003	1738-IB8M12	ArmorPoint 24 VDC Sink Input Module, 8 points, M12 Connectors	168.00	504.00
003	1738-AENT	ArmorPoint Communication Adaptor, EtherNet/IP, M12 Quick-Disconnect	540.00	1620.00
012	1738-IA2M12AC3	ArmorPoint 120 VAC Input Module, 2 points, AC Micro 3-pin Connector	170.00	2040.00
003	1738-FPD	ArmorPoint Field Power Distributor	105.00	315.00
		The pricing provided is List Price. Please consult your local sales person to get your specific pricing.		
		Total price, excluding network		15444.00

图 26-65 物料清单图

关闭 BOM 窗口。

28. 将所选的“系列”（Family）更改为 FLEX I/O，如图 26-66 所示。

IAB 现在会显示 FLEX I/O 的模块选择，如图 26-67 所示。



图 26-66　更改为 FLEX I/O

Module Selections:　☐ Substitute Combo/Self-Configuring Modules where appropriate

Module Type	Catalog Number	Points Requested	Module Information
Adapter Module			
Adapter Module	1794-AENT	N/A	Ethernet Adapter Module
Digital Input			
Digital AC Input, Non-Isolated	1794-IA16	24	85-132V AC Input Module, 16 Point
Digital DC Input, Sink	1794-IB16	24	24V DC Sink Input Module, 16 Point
Digital Output			
Digital AC Output, Non-Isolated	1794-OA8	24	85-132 VAC Output Module, 8 Point
Digital DC Output, Source, Non-Protected	1794-OB8	24	24V DC Source Output Module, 8 Point
Analog Input			
Analog Input, Current, Non-Isolated	1794-IE8	25	Analog Input Module, 8 Point
Analog Output			
Analog Output, Current, Non-Isolated	1794-OE4	25	Analog Output Module, 4 Point
Specialty Modules		Quantity	
2-Channel Incremental Encoder Input Module			

图 26-67　FLEX I/O 的模块选择

单击“完成”（Finish）。IAB 会向系统视图添加 DIO 向导图标，并再次显示，如图 26-68 所示对话框。

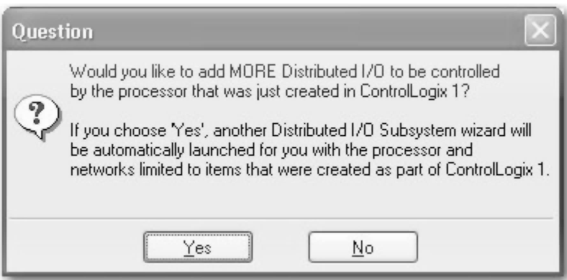


图 26-68　添加 DIO 向导

29. 单击“是”（Yes）。IAB 会再次启动 DIO 向导。浏览向导，以进行下列选择：

1）网络生成：使用现有网络 DeviceNet001，如图 26-69 所示。

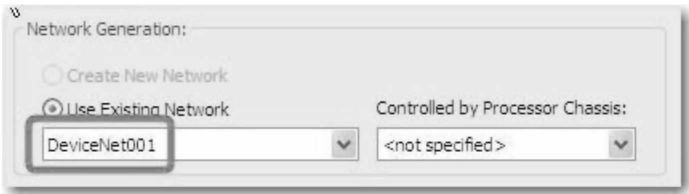


图 26-69　DeviceNet001 网络

2）50 个非隔离、AC 120V 和 50 个拉出型、DC 24V 数字量输入，如图 26-70 所示。

图 26-70　数字量设置图

3）50 个非隔离、AC 120V 数字量输出，如图 26-71 所示。

图 26-71　数字量设置图

4）无模拟量 I/O。

5）POINT I/O 系列和使用与处理器之间的机架优化连接，如图 26-72 所示。

单击“完成”（Finish）结束向导组态。IAB 会再次询问：是否要添加 DIO，如图 26-73 所示。

图 26-72　使用与处理器之间的机架优化连接

图 26-73　结束向导组态

30. 单击“否”（No）结束 DIO 创建循环，如图 26-74 所示。

可以根据需要多次进入 DIO 向导，以满足项目的需求。

在系统视图中，IAB 会显示一个 ControlLogix 子系统图标以及两个 DIO 子系统图标，以符合所创建的内容，如图 26-75 所示。

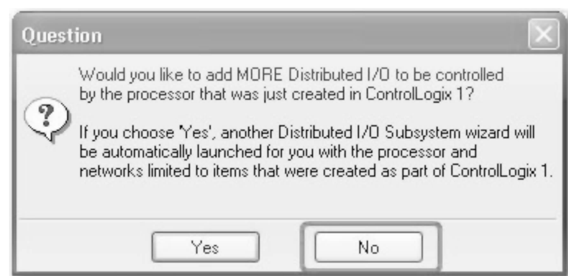


图 26-74　结束向导组态

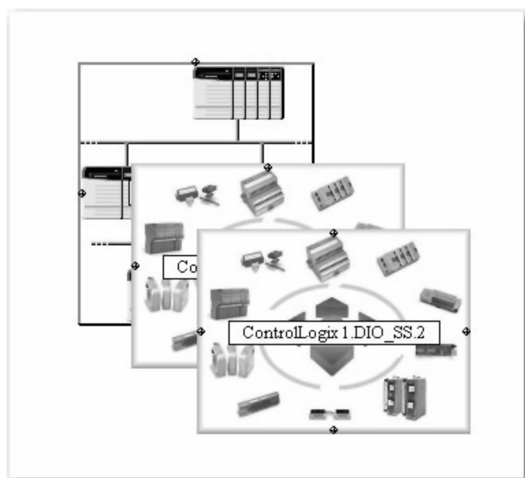


图 26-75　系统视图

31. 在系统视图中，右键单击 ControlLogix 向导图标并选择“查看子系统”（View Subsystem），如图 26-76 所示。

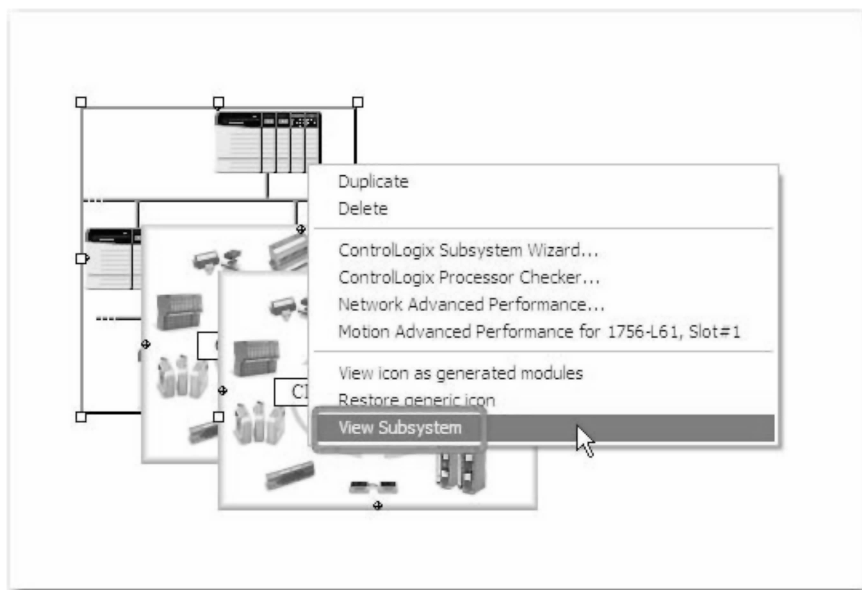


图 26-76　系统视图

IAB 会打开 ControlLogix 子系统的网络视图。可以看到 ControlNet 上存在六个 ControlLogix 机架，如图 26-77 所示。

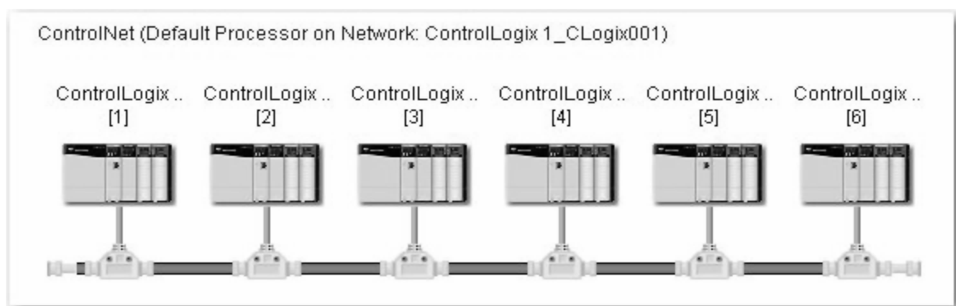


图 26-77 网络视图

单击“系统”（System）选项卡返回系统视图。右键单击其中一个 DIO 子系统图标，以查看向导创建的内容，然后对其他 DIO 子系统重复此操作，如图 26-78 所示。

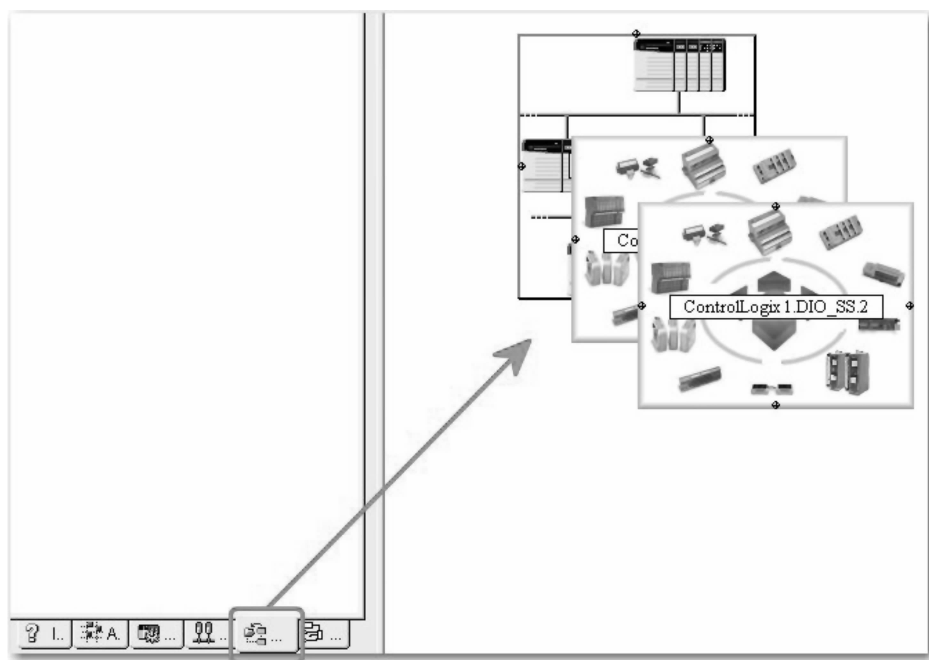


图 26-78 DIO 子系统图标

将看到 DIO 子系统的网络视图，如图 26-79 和图 26-80 所示。

关于子系统向导的部分到此结束。已使用 ControlLogix、DIO 子系统向导和工作流，创建了控制器的完整组态和网络上的分布式 I/O。

可以随时返回子系统向导更改组态，且仅需双击系统视图中的子系统图标即可生成组态。

CompactLogix 子系统向导的工作流与 ControlLogix 子系统向导的工作流非常相似，但唯一不同的是，前者是为 CompactLogix 平台定制的。有关 CompactLogix 子系统向导的详细信息，请参见 CompactLogix 实验练习。

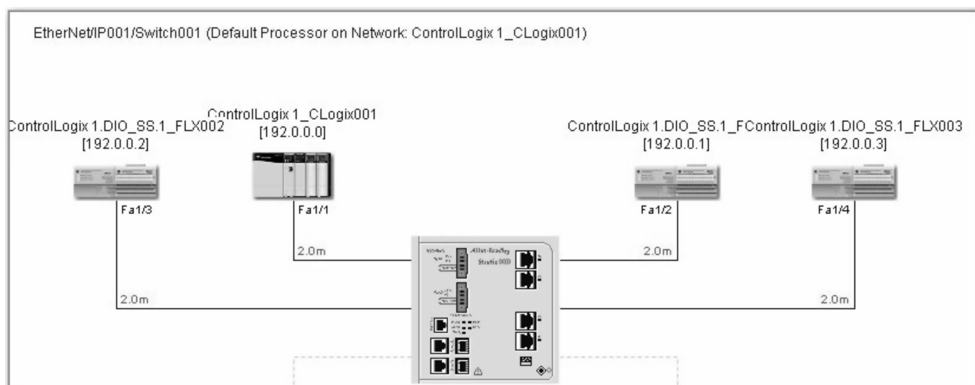


图 26-79　DIO 子系统的网络视图

四、硬件树和机架视图

本部分将介绍如何使用 IAB 中的硬件树来查看已组态的机架和设备。还会了解到许多在机架和硬件视图中可用的右键单击菜单选项。

1. 单击“硬件”（Hardware）选项卡并查看硬件树。硬件树是包含该 IAB 项目中创建的所有硬件的列表。现在，它会显示 ControlLogix 和进入的 DIO 向导所创建的全部内容，如图 26-81 所示。

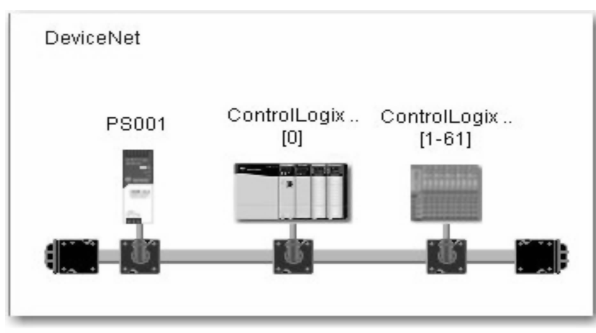


图 26-80　DIO 子系统的设备网络

列表中的第一项（默认选择）是第一个 ControlLogix 机架，其中包含处理器、通信模块（ControlNet、DeviceNet、EtherNet/IP 和 EWEB）以及 SERCOS 运动控制模块。还应注意 20% 的空闲插槽。

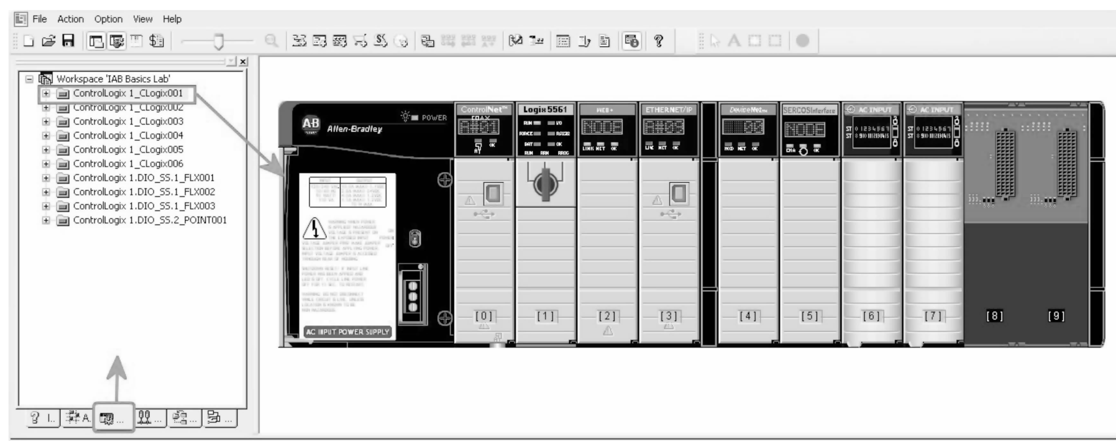


图 26-81　硬件树

2. 单击硬件树中第一个项目旁的加号（ControlLogix 1_CLogix001）。IAB 会显示 1756 机架。展开该机架，查看其中的模块。右键单击列表中的各项以查看可用选项，如图 26-82 所示。

3. 单击硬件树中的其他项，以查看所显示的内容，然后返回第一个机架视图。

4. 可以通过拖放的方式在机架内移动模块。单击机架中的最后一个模块，如图 26-83 所示。按住鼠标按键，将该模块拖动到机架中的最后一个插槽，如图 26-84 所示。

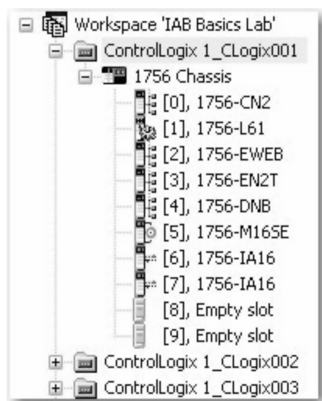


图 26-82　机架中的模块

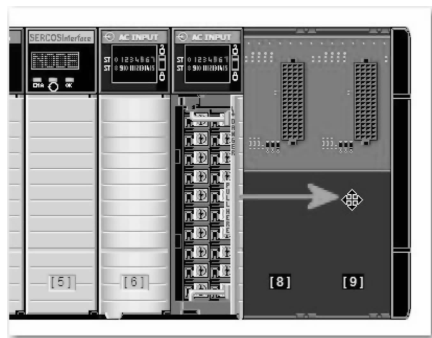


图 26-83　选中机架内模块

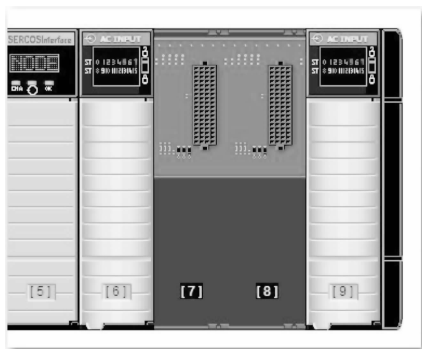


图 26-84　拖动至目标插槽



图 26-85　组态机架

5. 右键单击机架电源或机架中的空插槽，然后选择“组态机架”（Configure Chassis），如图 26-85 所示。

IAB 会显示“组态机架”（Configure Chassis）对话框，如图 26-86 所示。

6. 可以通过“组态机架”（Configure Chassis）对话框为每个机架分别设置选项。在“ControlLogix 组态机架”（ControlLogix Configure Chassis）对话框中，可以设置机架大小、接线方法、电源和一些高级性能数据（机架优化连接、额外内存、任务数）。在“组态机架”（Configure Chassis）对话框中进行的设置会覆盖在向导中进行的设置。

在该对话框中，选择 1756-PAR2 冗余电源，如图 26-87 所示。

单击“确定”（OK），并注意机架视图是如何变化以反映和所做的更改的，如图 26-88 所示。

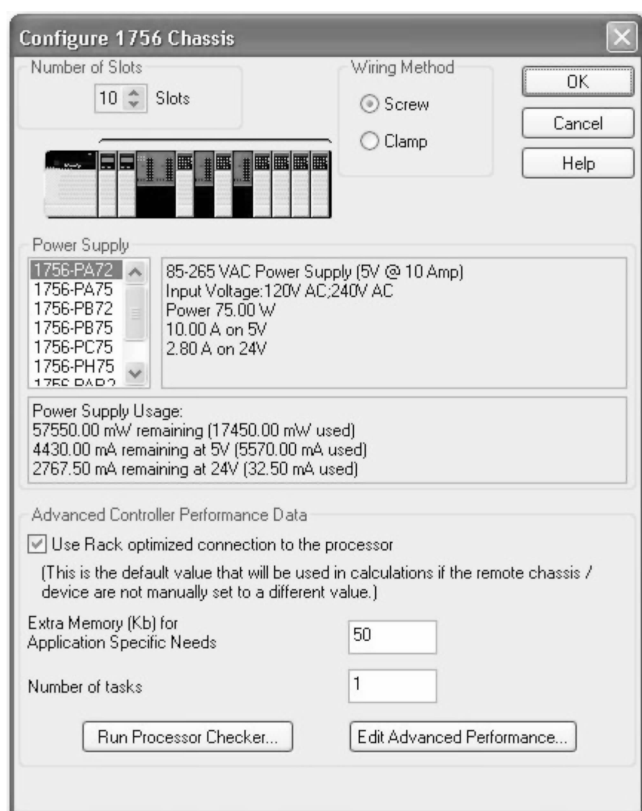


图 26-86　组态机架对话框

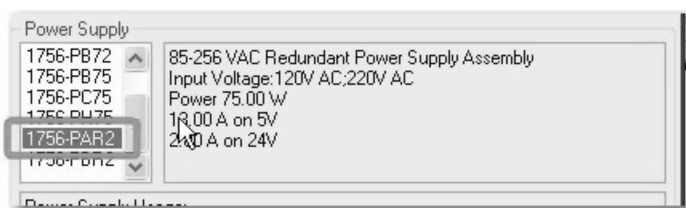


图 26-87　电源选择

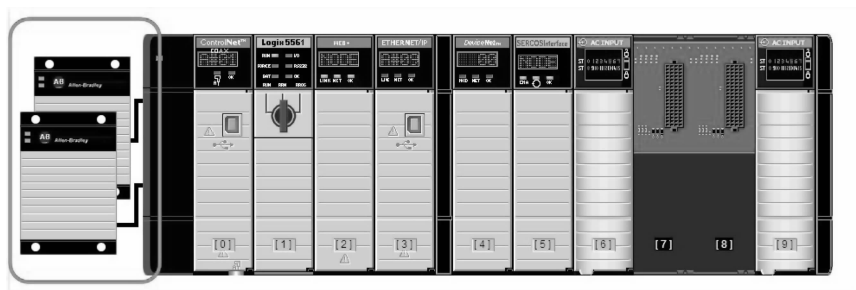


图 26-88　电源选择后变化图

7. 在硬件树中，单击其他 ControlLogix 机架，以查看其组态是否未发生变化。还应单击 FLEX I/O，如图 26-89 所示。和 POINTI/O 机架，如图 26-90 所示。并查看其“组态机架”（Configure Chassis）对话框，可看到这些重要对话框中可用的不同组态选项。



图 26-89　FLEX I/O 机架

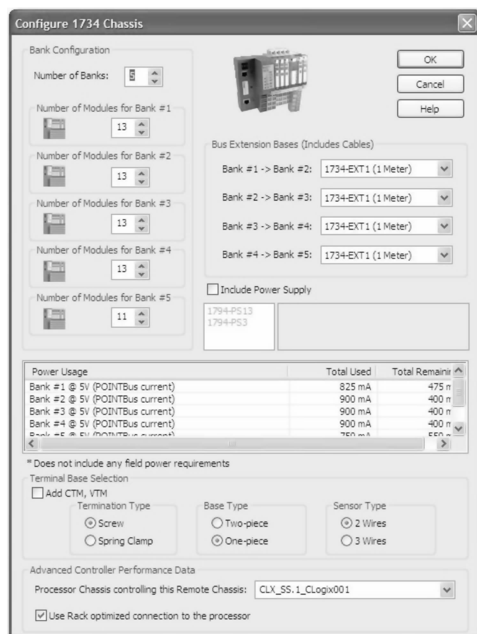


图 26-90　POINT I/O 机架

8. 在机架视图中，右键单击每一个模块。将看到，每个模块都具有包含各种选项的右键单击菜单，而这些选项适用于单独的模块。通过右键单击菜单，可以：

1）删除模块；

2）打开关于模块的产品文档；

3）将模块连接到网络，或断开模块的网络连接；

4）打开连接到网络的模块的“通道属性”（Channel Properties）对话框；

5）打开对话框，将模块附件添加到物料清单；

6）对于处理器，运行处理器检测程序，以检查内存和连接使用情况；

7）打开网络高级性能对话框；

8）对于运动控制模块，打开运动控制向导，对运动控制进行组态（请参见运动控制实验练习）；

9）对于 I/O 模块，打开机旁布线向导，以组态接线系统模块、分线盒、机旁电缆和终端设备（请参见机旁实验练习）；

10）可以随意插入空模块；

11）可以卸下空插槽；

12）在 FlexI/O 的机架中为每个模块编辑端子底座。

五、网络视图

单击“网络”（Network）选项卡。IAB 会显示网络视图。按照图 26-91 所示展开 ControlLogix 1_Cnet ControlNet 网络。请注意，可以增大左侧导航窗口的宽度，以显示其中的更多内容。请注

意，网络选项卡显示的是，使用向导创建的 ControlNet、EtherNet/IP 和 DeviceNet 三个网络。

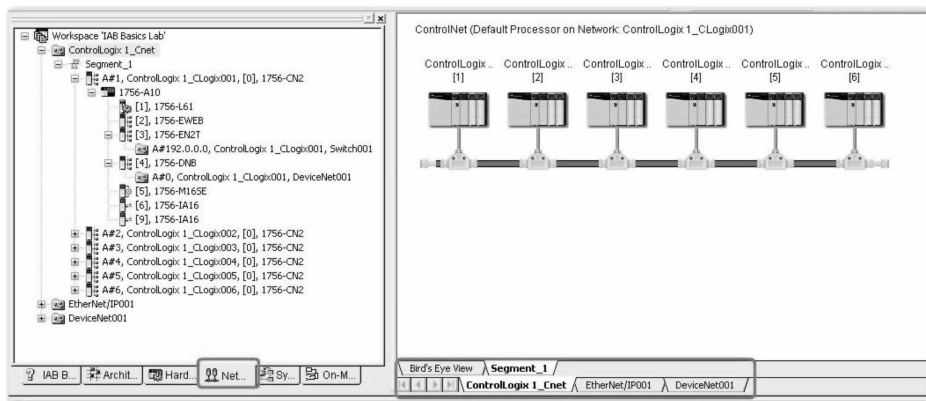


图 26-91　向导创建的三个网络

请注意，每个网络的网络视图中都存在一个选项卡。单击该网络选项卡，显示每个网络视图。还可以在网络树中单击网络名称。还应注意，对于每个网络来说，都存在一个或多个子选项卡。

六、将通信模块连接到新网络

请记住，在 ControlLogix 向导中指定的 1756-EWEB 模块并未通过向导连接到任何网络。现在，想要将该模块连接到一个 EtherNet/IP 网络，但并不是其他 EtherNet/IP 模块连接到的同一网络。

单击“硬件”（Hardware）选项卡。IAB 会显示包含 EWEB 模块的 ControlLogix 机架的机架视图。右键单击 EWEB 模块并选择“将通道‘.’连接到新 EtherNet/IP 网络 > 星型”（Connect channel‘.’to a new EtherNet/IP Network > Star），如图 26-92 所示。

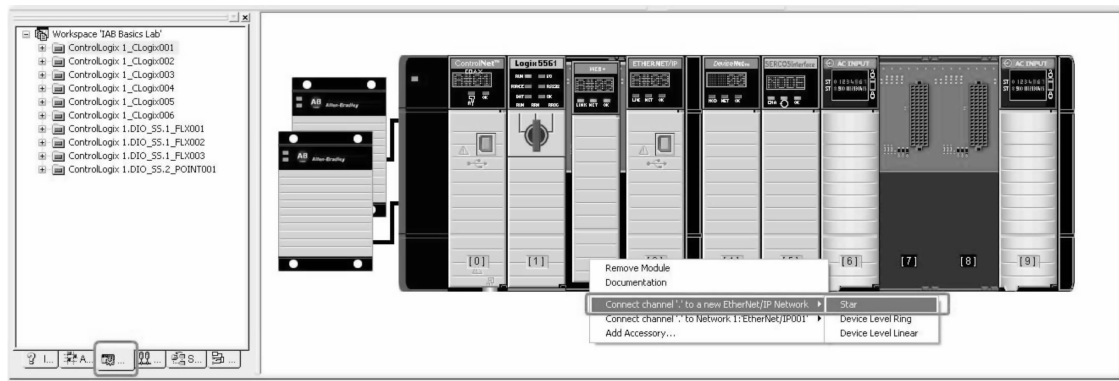


图 26-92　连接到新网络

IAB 会显示“新网络”（New Network）对话框。键入名称 Supervisory Ethernet，然后单击“确定”（OK），如图 26-93 所示。

IAB 会要求提供 IP 地址。在此处指定的地址仅供参考。对于本练习来说，仅需通过单击“确定”（OK）接受默认设置即可，如图 26-94 所示。

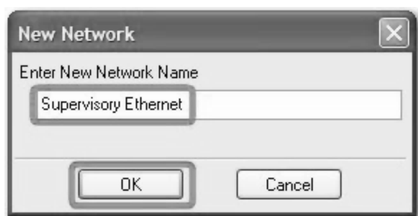


图 26-93 设定新网络名

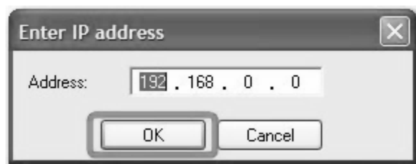


图 26-94 设置 IP 地址

IAB 会显示以下消息，该消息会通知：已将安装有 EWEB 模块的机架中的处理器设置为该网络的默认处理器。该消息与性能检测相关，稍后会在本试验中介绍该方面的内容。

单击“继续”（Continue），如图 26-95 所示。

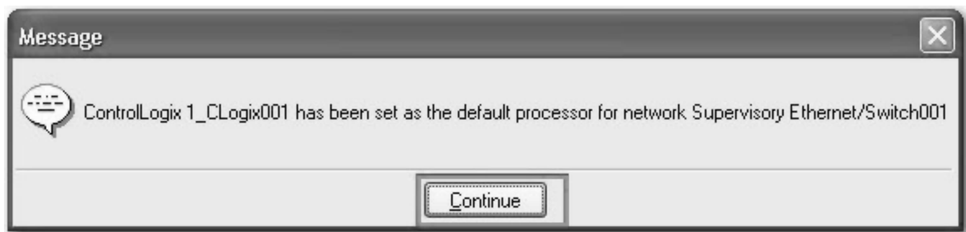


图 26-95 继续按钮

IAB 会显示新创建的网络的网络视图。请注意网络的新增选项卡、网络树中的新增条目，以及连接到网络交换机的 ControlLogix 机架，如图 26-96 所示。

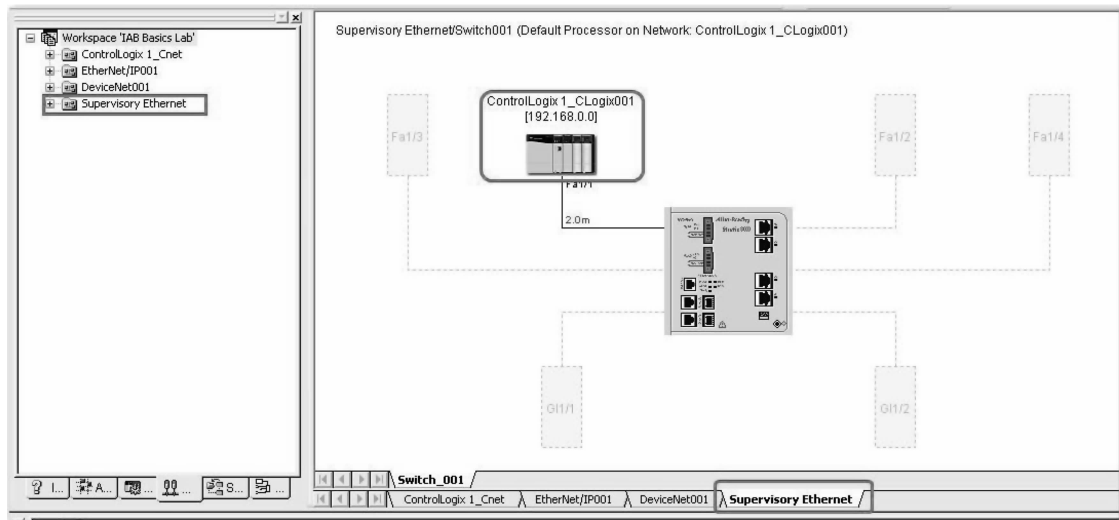


图 26-96 新建网络的视图

七、手动添加网络和硬件

1. 添加 DeviceNet 网络：

1）单击“新网络”（New Network）按钮，如图 26-97 所示。该操作与执行时显示哪一视图无关。

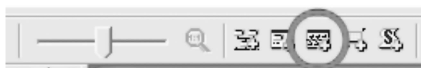


图 26-97　新网络图标

将弹出“选择网络…”（Choose a Network…）对话框，如图 26-98 所示。

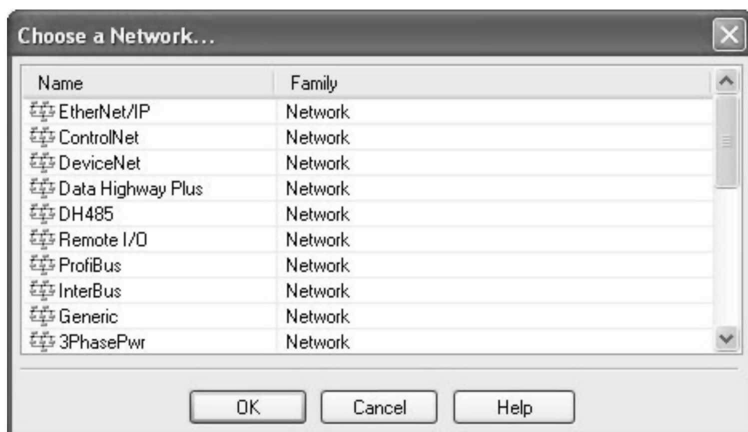


图 26-98　选择网络对话框

2）选择 DeviceNet，然后单击“确定”（OK），如图 26-99 所示。

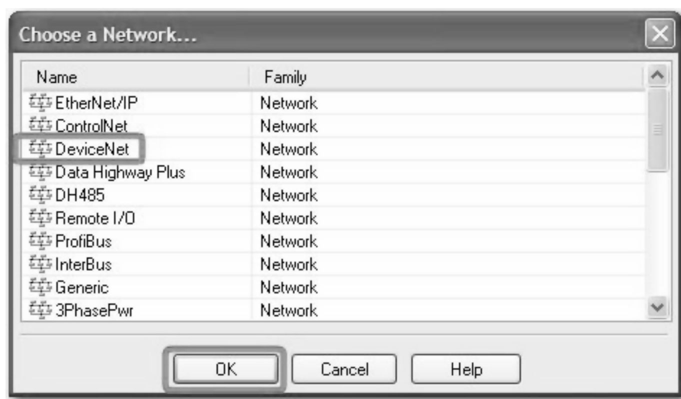


图 26-99　选择 Device Net 图

将会弹出“新网络”（New Network）框，该框包含系统为新网络创建的名称以及所选的默认电源，如图 26-100 所示。

3）键入名称 My DeviceNet，然后单击“确定”（OK）。

将出现包含所选默认电源的新 DeviceNet 网络（见图 26-101）。请注意，存在一个用于网络的选项卡和一个用于干线的选项卡。

如果向 DeviceNet 添加分支拓扑，IAB 将为每个分支添加选项卡，如图 26-101 所示。

2. 将 ArmorPoint 适配器添加到新的 DeviceNet 网络。

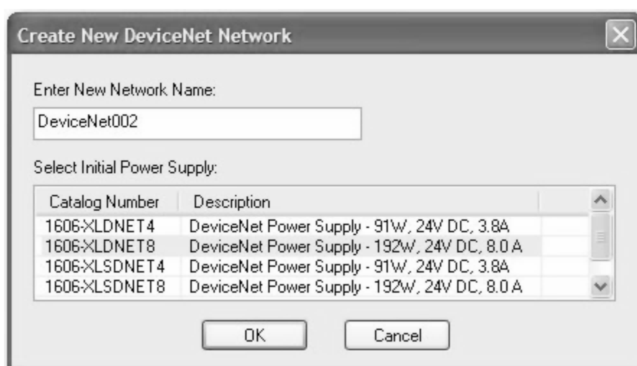


图 26-100　新网络框图

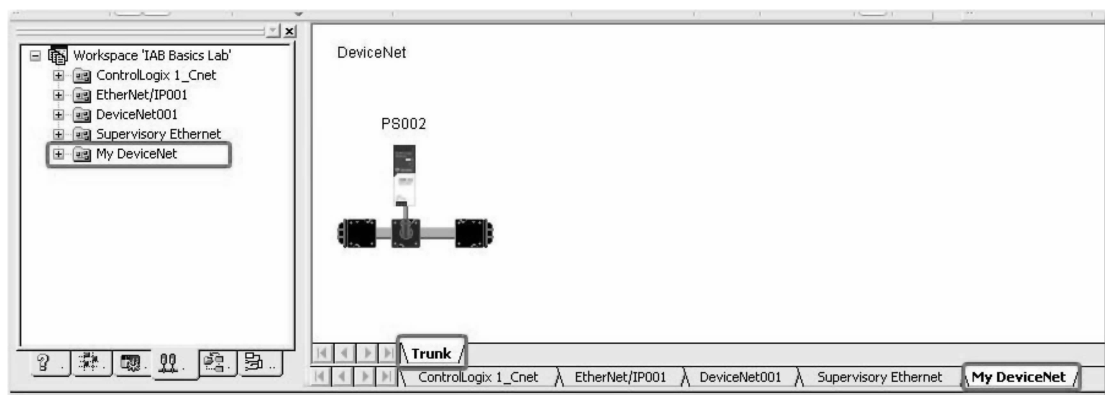


图 26-101　新 Device Net 网络图

IAB 窗口的底部存在“设备列表”（Device List）区域。该区域会列出可以通过拖放方式添加到网络和机架的设备，如图 26-102 所示。

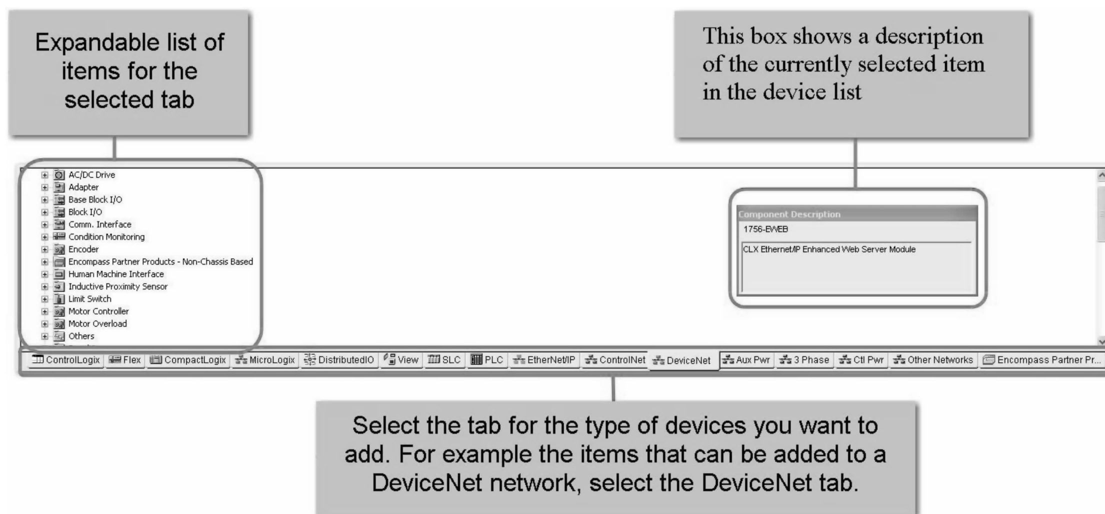


图 26-102　IAB 窗口的底部图

1）单击 DeviceNet 选项卡，然后展开设备列表中的“适配器”（Adapter）文件夹，如图 26-103 所示。将出现可以添加到 DeviceNet 网络的 I/O 适配器列表。

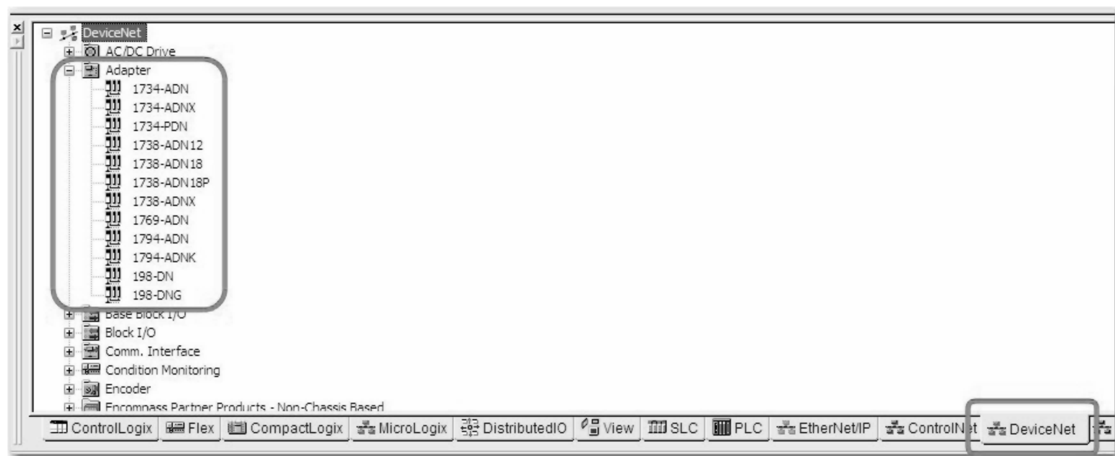


图 26-103　适配器文件夹

2）左键单击 1738-ADN18 适配器，然后按住鼠标按键，将其拖动到网络视图中。将出现一个加号图标，以显示该设备对于网络合法，如图 26-104 所示。

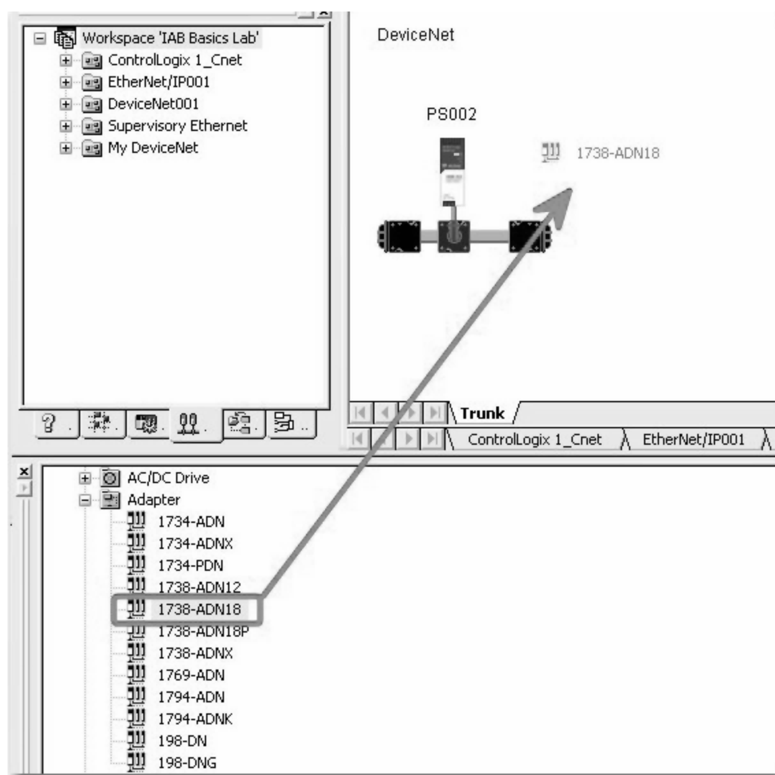


图 26-104　选择 1738-ADN18 适配器

释放鼠标按键后，ArmorPoint 节点会出现在网络上。请注意，IAB 会自动添加 DeviceNet 介质——扁平干线电缆（默认）和关联的分接器和支线电缆，如图 26-105 所示。

以下是在 IAB 中采用拖放方式添加设备的基本步骤。

在一些视图中，将模块拖动到特定位置以将其成功添加是非常重要的。例如，在机架视图中，必须将模块拖动到特定的插槽。在可能具有多条分支的电源网络视图中，必须将项目拖动到要将其添加至的特定分支。

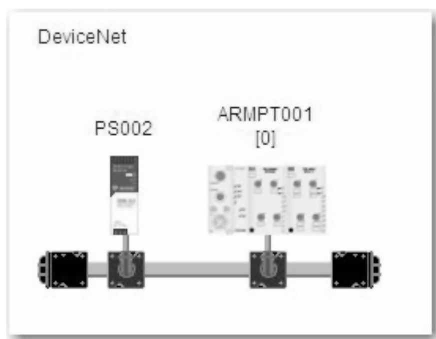


图 26-105　选择节点

3. 单击“硬件”（Hardware）选项卡。刚刚添加的 ArmorPoint 适配器会在列表底部显示为 ARMPT001。单击 ARMPT001 项（还可以通过双击网络视图中的节点图标打开硬件视图）。将出现 ArmorPoint 节点的机架视图。1738-ADN18 适配器将显示，其中具有三个空插槽。IAB 会在通过拖放方式将适配器添加到网络时，自动创建该 ArmorPoint 组件（“机架”），以及与 DeviceNet 网络之间的连接，如图 26-106 所示。

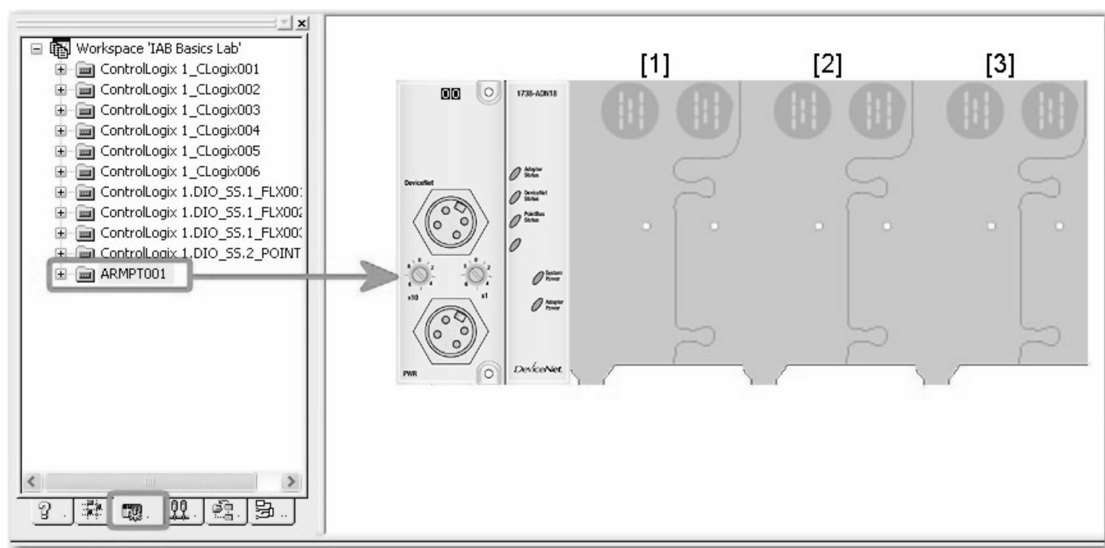


图 26-106　节点的机架视图

4. 在进行下一步操作之前，请单击“保存”（Save）按钮以保存到目前为止所执行的操作，如图 26-107 所示。



图 26-107　单击保存

5. 添加 I/O 模块。

单击“设备列表”(Device List）窗口中的“分布式 I/O”（Distributed I/O）选项卡。“设备列表”（Device List）现在会显示不同的分布式 I/O 系列中的 I/O 模块（包括 ArmorPoint），如图 26-108 所示。

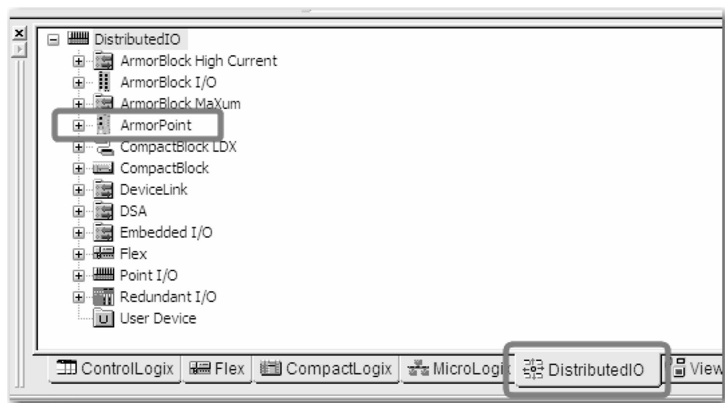


图 26-108 分布式 I/O 选项卡

单击 ArmorPoint 文件夹旁边的加号以展开文件夹中的内容。然后，展开“I/O 模块”（I/O Module）文件夹，接着展开“模拟量”（Analog）和“数字量”（Digital）文件夹。将列出 ArmorPointI/O 模块。

将以下 I/O 模块从分布式 I/O 设备列表拖放到 POINTI/O 硬件视图中。

1）第一个插槽：模拟量 I/O-1738-IT2IM12；

2）第二个插槽：数字量 I/O-1738-IA2M12AC4。

确保将模块拖动到空插槽中，以便硬件视图能够接受该模块。

请注意，当添加交流 I/O 模块时，IAB 会自动安装 1738-FPD 模块。以下是 IAB 是如何通过应用规则来帮助确保创建在技术上有效的组态的示例，如图 26-109 所示。

6. 组态机架。

在机架视图中，右键单击适配器模块并选择“组态机架”（Configure Chassis），如图 26-110 所示。

在出现的对话框中，将“组数目”（Number of Banks）增至 2。在第 2 组中指定两个插槽，然后在第 1 组和第 2 组之间选择 1738-EXT3 三米连接电缆。请注意，还可以在该对话框的底部查看关于耗电量的信息，并为该 I/O 机架指定控制器。单击“确定”（OK），如图 26-111 所示。

硬件视图改为显示第 2 组以及连接第 1 组和第 2 组的电缆。将 1738-IB4M12 模块拖放到第 2 组的空插槽中。将鼠标悬停在该视图中各项的上方以查看其相关信息，如图 26-112 所示。

右键单击空插槽并选择“删除空插槽”（Remove empty slot）。这样便可删除第 1 组中的空插槽，如图 26-113 所示。

至此便完成了 ArmorPoint 组态。接下来，要将 PowerFlex 变频器添加到 DeviceNet。

7. 在继续进行操作之前，请单击 IAB 工具栏上的“保存”（Save）按钮以保存之前的操作，如图 26-114 所示。

八、将 PowerFlex 变频器添加到 DeviceNet 网络（RAISE 配置器）

接下来，将使用 RAISE 配置器来选择要添加到 DeviceNet 的 PowerFlex 变频器。

注：可以使用此处介绍的相同步骤将 ArmorStart 电动机控制器添加到 DeviceNet 网络。

通过单击“网络”（Network）选项卡显示网络视图，然后单击列表中的“DeviceNet”（My DeviceNet）项，如图 26-115 所示。

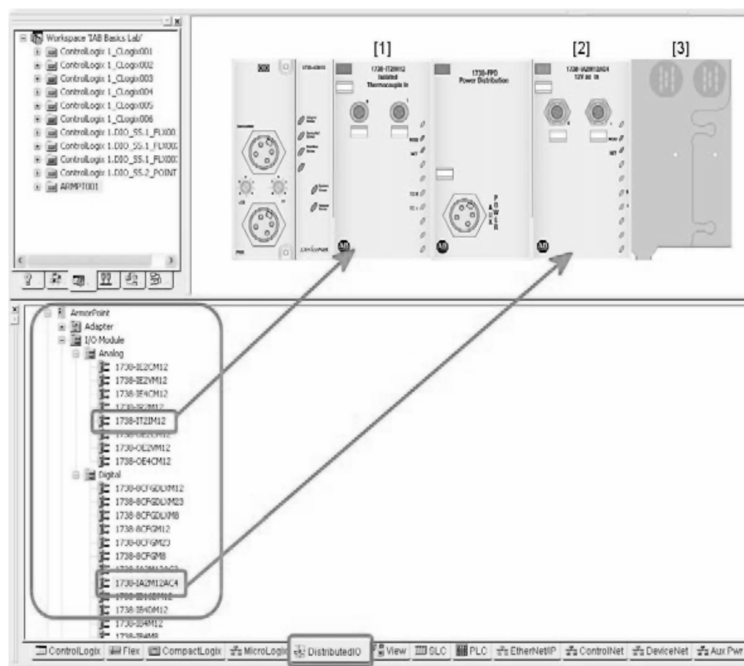


图 26-109　硬件视图

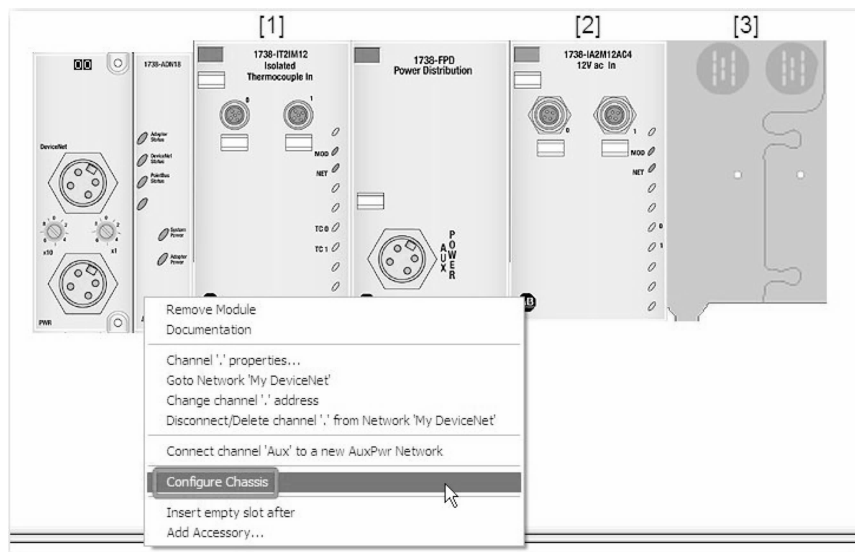


图 26-110　机架视图

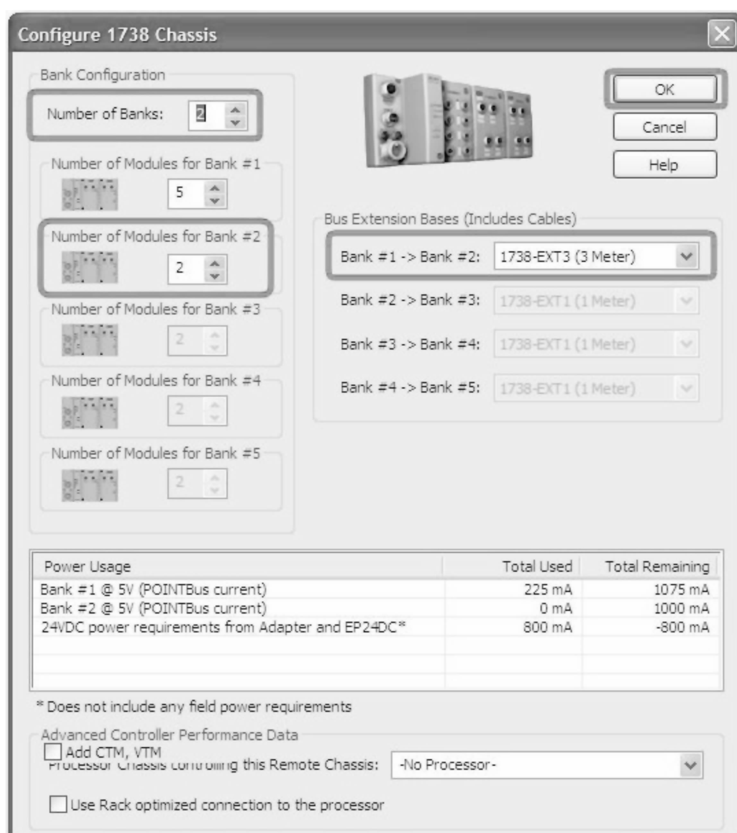


图 26-111　硬件组态视图

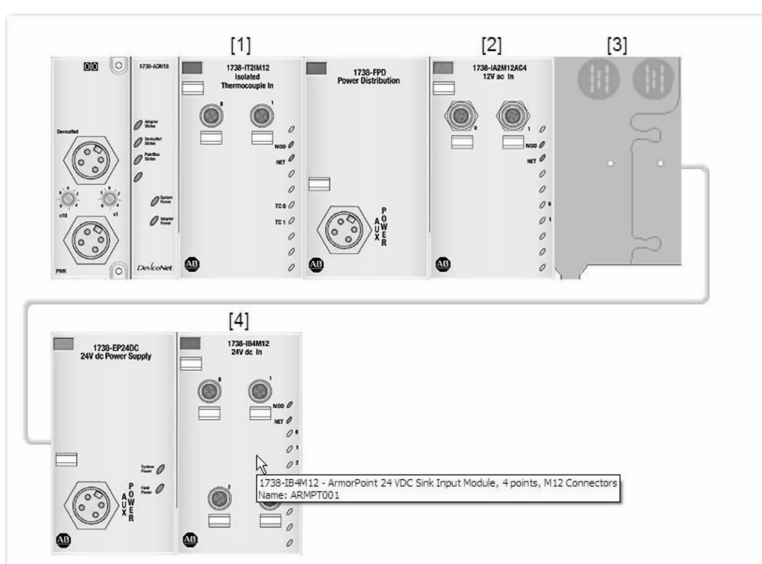


图 26-112　硬件更改

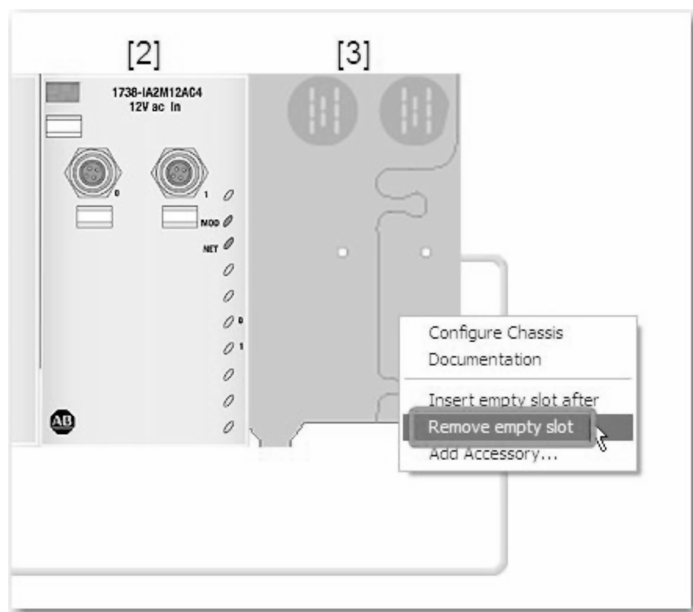


图 26-113 删除空插槽图



图 26-114 保存按钮

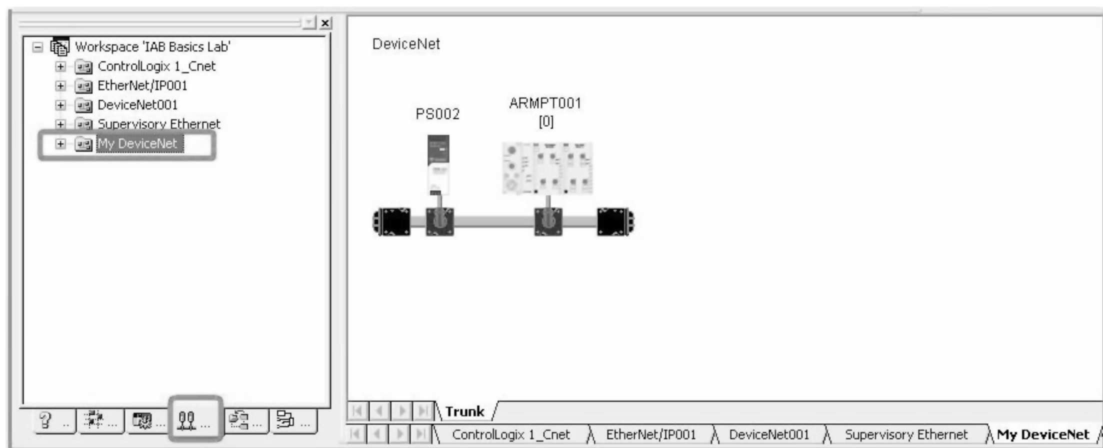


图 26-115 网络视图

1. 单击“DeviceNet”选项卡，然后向下滚动设备列表，直至看到 PowerFlex 400。展开 PowerFlex 400 文件夹，将 22C＊＊＊D＊＊＊ 拖动到 DeviceNet 网络视图中，如图 26-116 所示。

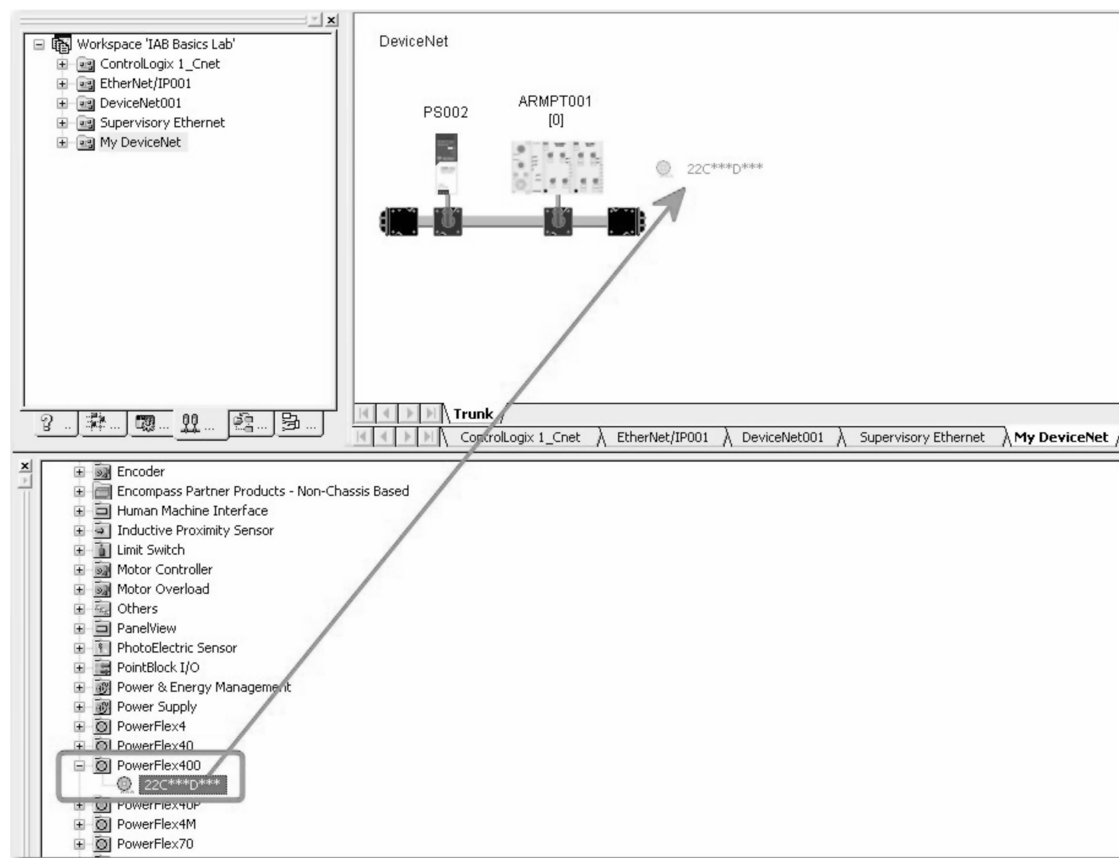


图 26-116　变频器配置图

2. 将 PowerFlex 变频器（或 ArmorStart 电动机控制器）添加到网络时，IAB 会自动为添加到网络中的项目启动 RAISE 配置器。花几分钟的时间来观察 RAISE 配置器界面的布局，如图 26-117 所示。

3. 单击“版本电压代码”（Version Voltage Code）。会在右侧面板中显示输入电压选项列表，如图 26-118 所示。

单击“480 VAC，3PH.”的单选按钮。所选的输入电压会显示在“输入电压”（Input Voltage）项旁边，如图 26-119 所示。

请注意，由于已将 DeviceNet 添加到 DeviceNet 网络，因此会自动为“网络类型”（Network Type）选择该变频器，如图 26-120 所示。

进行选择时，目录号中的问号将会用字母或对应于所选项的数字代替。

到达“附件”（Accessories）项后，请转至步骤 3。

4. 在“附件”（Accessories）部分中，单击“附件和备件”（Accessory Kits and Spare Parts），然后选择“软件套件”（Software Kits）。将显示红色问号和注意事项，告知该项目需要附加组态。单击所出现的红色问号中的任一个问号，如图 26-121 所示。

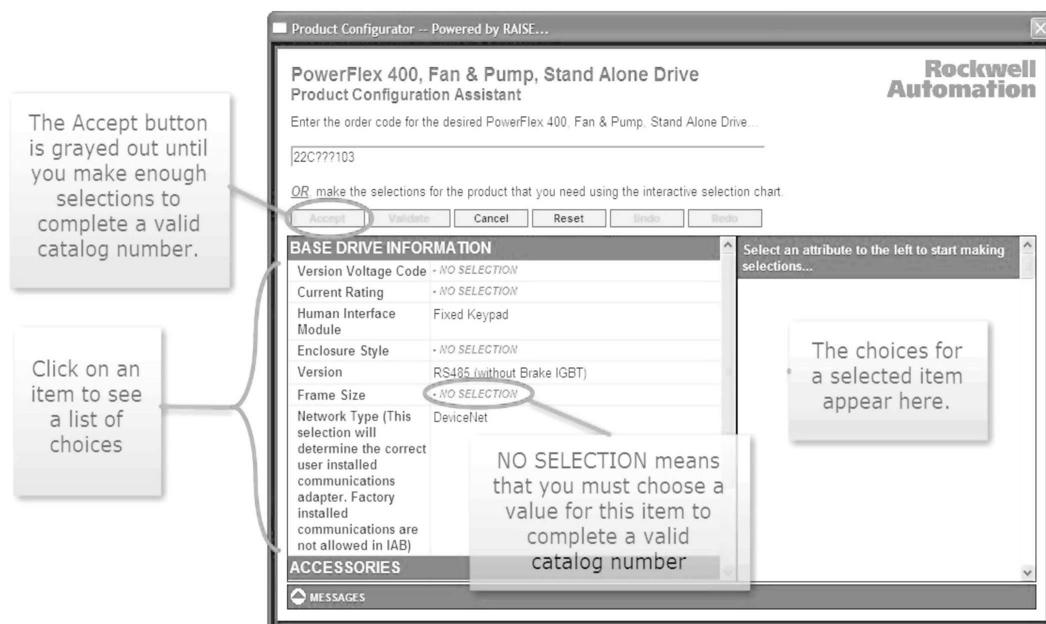


图 26-117　配置图



图 26-118　版本电压代码



图 26-119　版本电压代码设置

BASE DRIVE INFORMATION	
Version Voltage Code	480 VAC, 3 PH
Current Rating	10 Amps 5 HP
Human Interface Module	Fixed Keypad
Enclosure Style	IP20 (Open)
Version	RS485 (without Brake IGBT)
Frame Size	Frame Size C
Network Type (This selection will determine the correct user installed communications adapter. Factory installed communications are not allowed in IAB)	DeviceNet

图 26-120　网络类型图

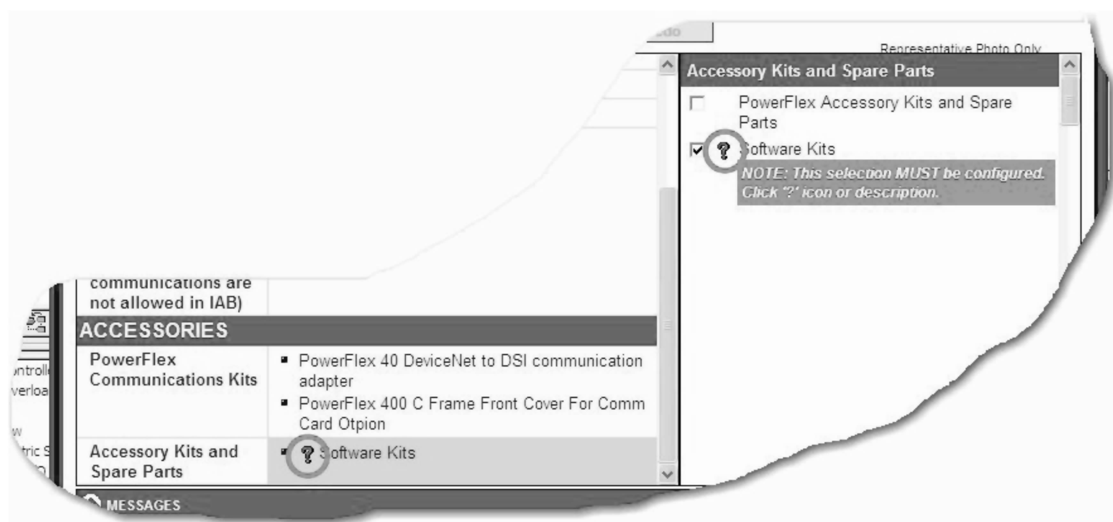


图 26-121　软件套件选择

将为变频器软件显示另一个配置器窗口。

5. 单击“软件”（Software）项。选择列表中的第一个选项（Drive Explorer），然后单击“接受”（Accept）按钮，如图 26-122 所示。

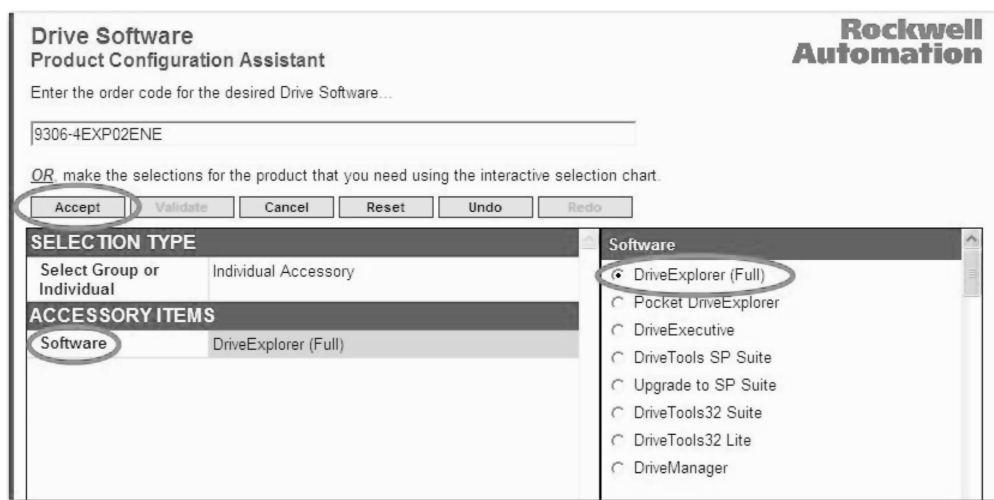


图 26-122　软件设置图

6. 原始 PowerFlex 400 配置器窗口会再次出现。请注意“附件和备件”（Accessory Kits and Spare Parts）项旁边的绿色复选标记。此标记指示该项目的选项已组态。单击“接受”（Accept）按钮完成组态，如图 26-123 所示。

在 IAB 网络视图中，PowerFlex 变频器会出现在 DeviceNet 网络上。

7. 单击“快速 BOM”（Quick BOM）按钮，查看 DeviceNet 的 BOM（当前窗口中正在显示的内容）。确认 PowerFlex 项已添加到 BOM。在继续进行操作之前，单击“关闭”（Close）以关闭

BOM，如图 26-124 所示。

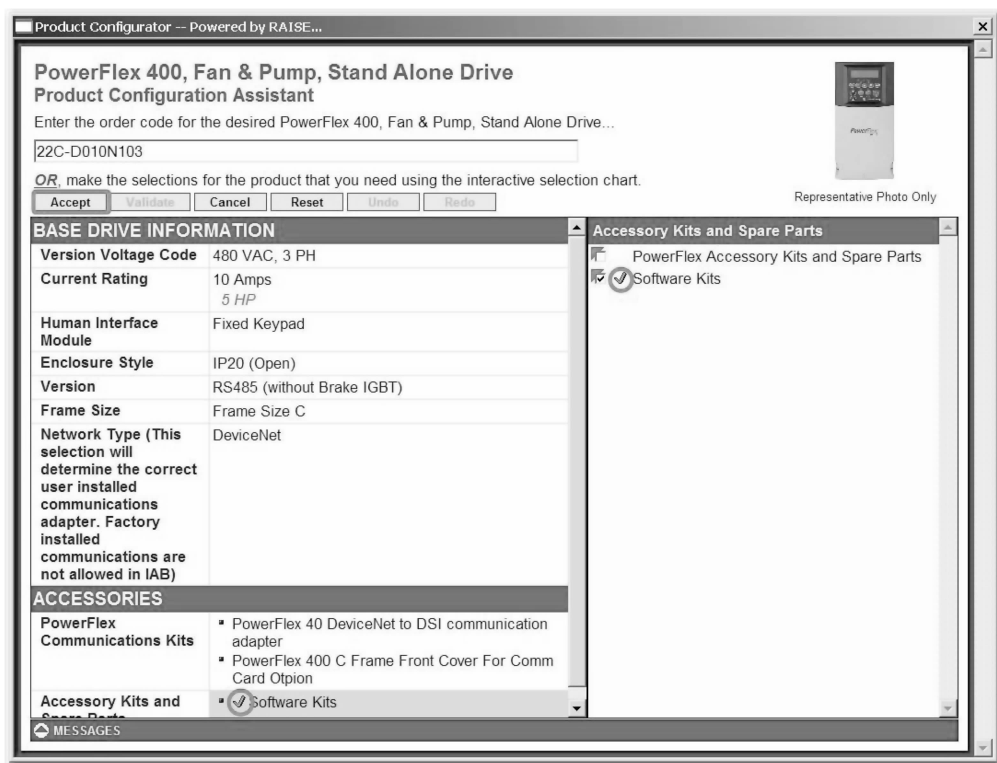
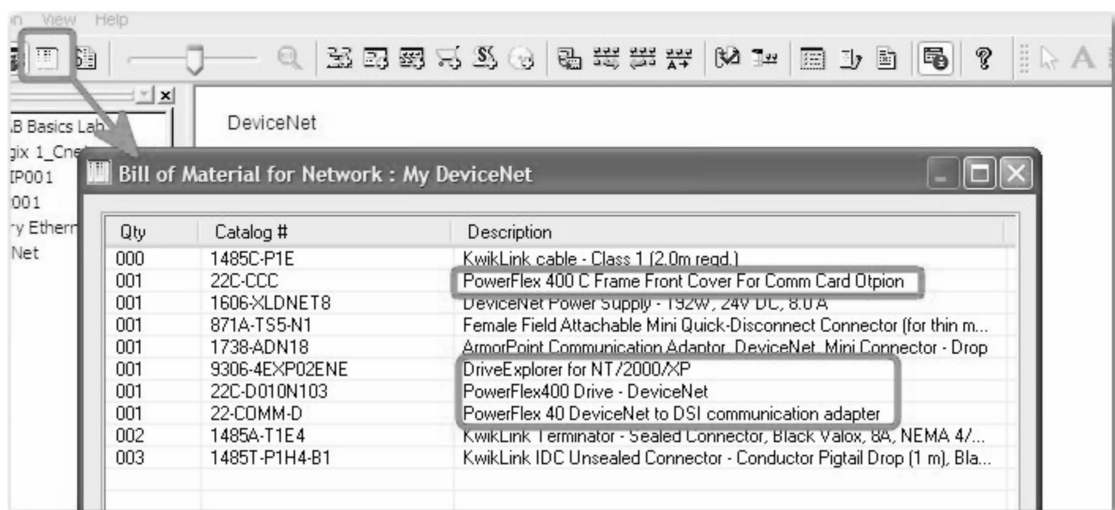


图 26-123　配置器组态



Qty	Catalog #	Description
000	1485C-P1E	KwikLink cable - Class 1 (2.0m reqd.)
001	22C-CCC	PowerFlex 400 C Frame Front Cover For Comm Card Otpion
001	1606-XLDNET8	DeviceNet Power Supply - 192W, 24V DC, 8.0 A
001	871A-TS5-N1	Female Field Attachable Mini Quick-Disconnect Connector (for thin m...
001	1738-ADN18	ArmorPoint Communication Adaptor, DeviceNet, Mini Connector - Drop
001	9306-4EXP02ENE	DriveExplorer for NT/2000/XP
001	22C-D010N103	PowerFlex400 Drive - DeviceNet
001	22-COMM-D	PowerFlex 40 DeviceNet to DSI communication adapter
002	1485A-T1E4	KwikLink Terminator - Sealed Connector, Black Valox, 8A, NEMA 4/...
003	1485T-P1H4-B1	KwikLink IDC Unsealed Connector - Conductor Pigtail Drop (1 m), Bla...

图 26-124　BOM 图

8. 将光标移至网络视图中的变频器图像上，并注意在所出现的工具提示中显示的变频器的目录号，如图 26-125 所示。

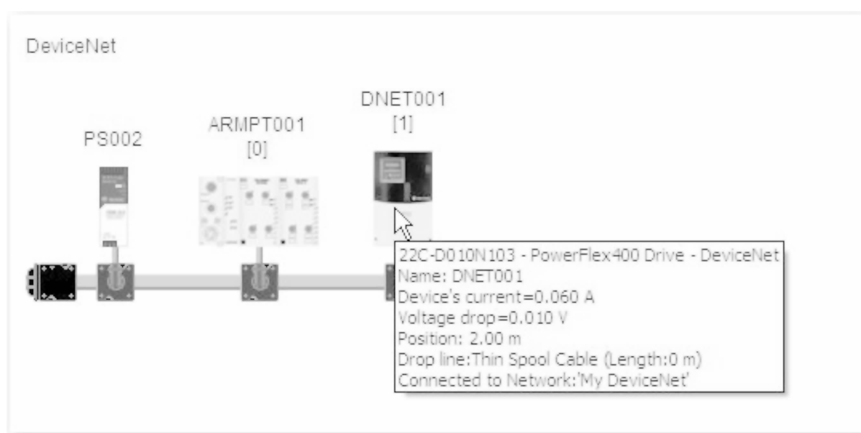


图 26-125　变频器目录号

在“网络列表”（Network List）中，展开“ 的 DeviceNet”（My DeviceNet）项，并注意变频器及其目录号将被列出，如图 26-126 所示。

9. 在继续进行操作之前，请单击 IAB 工具栏上的“保存”（Save）按钮以保存之前的操作，如图 26-127 所示。

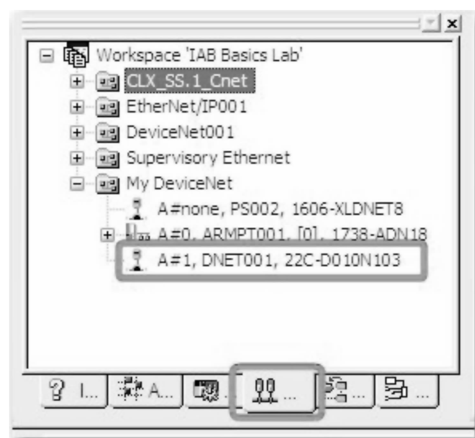


图 26-126　变频器目录号

九、将 PanelView 添加到以太网（新硬件）

通过拖放的方式，将设备添加到特定网络是一种非常便捷的方法。此外，还可以使用新硬件功能将设备添加到组态中。使用新硬件功能时，所选设备不会自动连接到任何网络；将必须在添加该设备后手动将其连接到网络。

步骤

1. 单击“新硬件”(New Hardware）按钮。(还可以从“操作”（Action）菜单中选择“添加新硬件”（Add New Hardware)），如图 26-128 所示。



图 26-127　保存按钮



图 26-128　添加新硬件

将出现“选择设备…”（Choose a device…）对话框，如图 26-129 所示。

2. 单击“视图”（View）的单选按钮，然后单击同一行中的“选择”（Choose）按钮，如图 26-130 所示。

将出现“查看”(View）产品的“选择设备…”(Select a device…）对话框，如图 26-131 所示。

3. 展开 PanelView Plus 文件夹，然后展开 PVP700 文件夹并选择目录号 2711P-B7C4A1。注意对话框底部的所选目录号的产品说明。单击“确定”（OK），如图 26-132 所示。

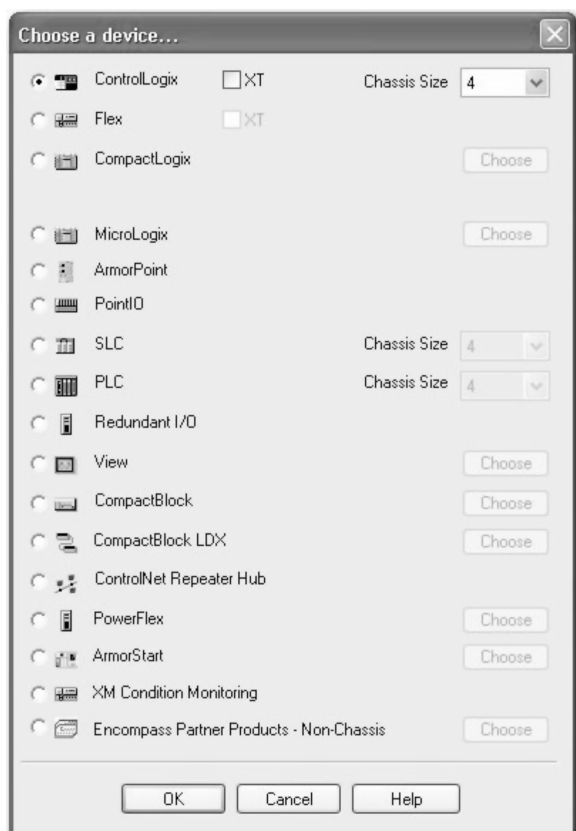


图 26-129　选择设备

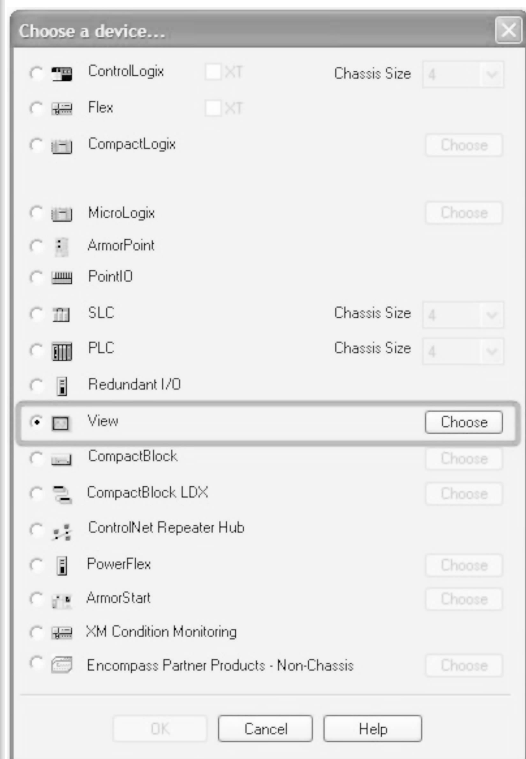


图 26-130　视图选择

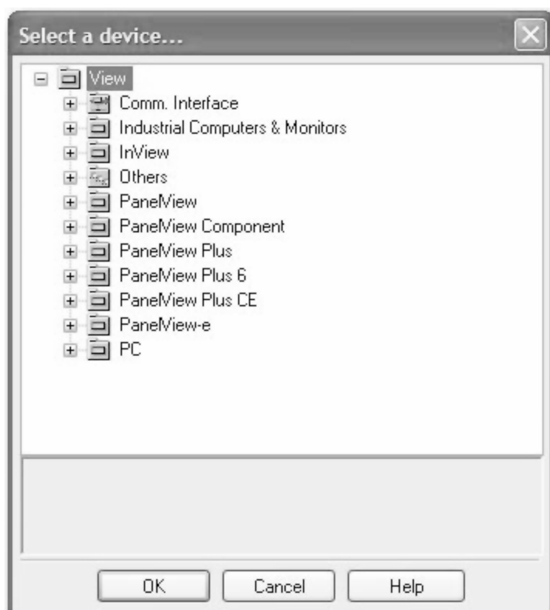


图 26-131　选择设备

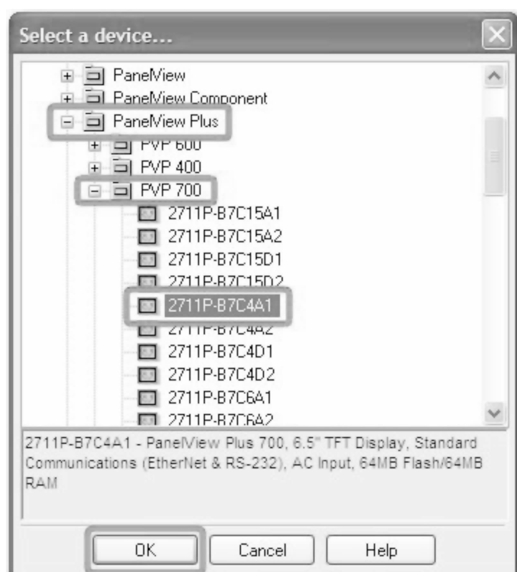


图 26-132　选择目录

将在 IAB 中打开所选设备的硬件视图，并且设备会添加到硬件列表中，如图 26-133 所示。

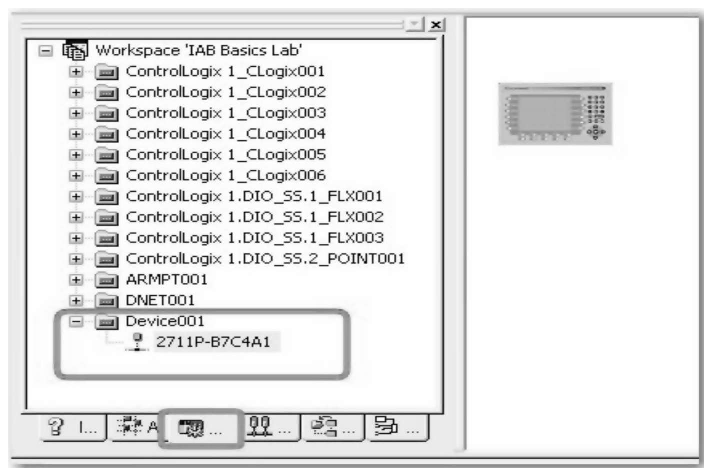


图 26-133　硬件视图

单击“网络”（Networks）选项卡，然后选择所有网络视图，以验证 PanelView Plus 未连接到任何网络。

4. 单击“硬件”(Hardware）选项卡，然后右键单击 Device001 项并选择“重命名设备”（Rename Device)，如图 26-134 所示。键入名称 PanelViewPlus，然后点击“回车”（Enter），如图 26-135 所示。

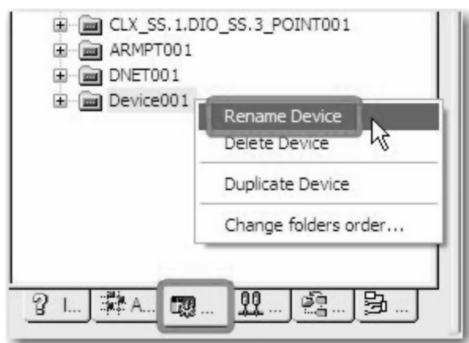


图 26-134　重命名设备

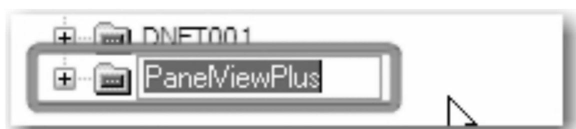


图 26-135　回车确认

5. 单击“硬件”(Hardware）选项卡，然后单击 PanelViewPlus 项，以再次打开 PanelView-Plus 终端的硬件视图，如图 26-136 所示。

6. 右键单击 PanelView Plus 图像并选择“将通道‘.’连接到网络 2：‘Supervisory Ethernet’”(Connect channel ‘.’ To Network 2：’ Supervisory Ethernet’)，然后将光标移动到右侧并选择 Switch001，如图 26-137 所示。

7. 再次打开“网络视图”（Network View）（单击“网络”（Network）选项卡并选择 Supervisory Ethernet)，并看到 PanelView Plus 已连接，如图 26-138 所示。

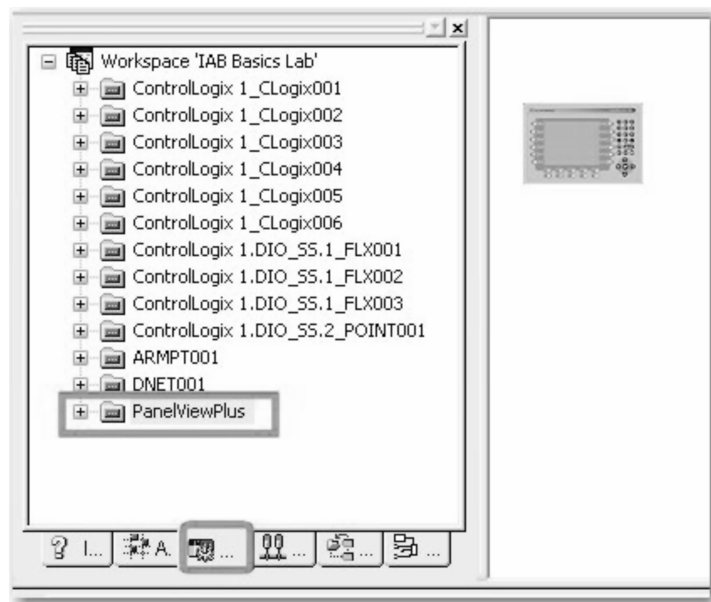


图 26-136　硬件视图



图 26-137　通道设置图

8. 在继续进行操作之前，请单击 IAB 工具栏上的“保存”（Save）按钮以保存之前的操作，如图 26-139 所示。

十、组态网络

目标

将设备添加到 DeviceNet、ControlNet 或 EtherNet/IP 网络时，IAB 会根据指定的默认值自动选择所需的电缆、分接器和支线电缆。本练习会介绍如何使用 DeviceNet 向导更改 DeviceNet 组态的默认值。

在 IAB 中，存在 ControlNet 和 DeviceNet 的网络向导。不存在受 IAB 支持的其他网络的向导。对于以太网网络，可以通过交换机属性对话框（右键单击交换机并选择“组态接线”（Configure Cabling））指定物理介质。

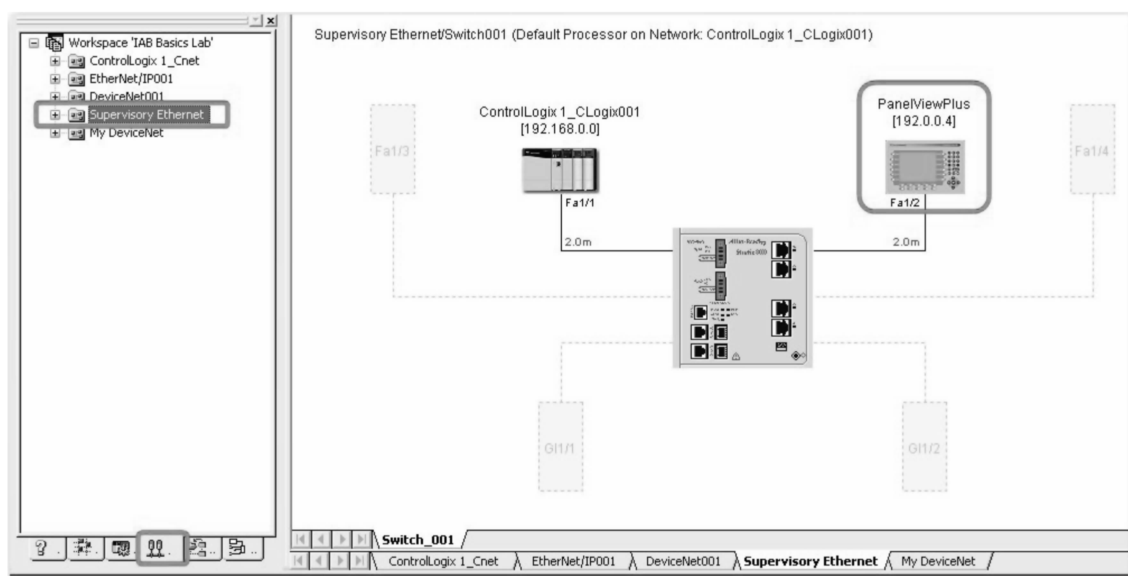


图 26-138　网络视图

步骤



图 26-139　保存按钮

1. 单击“网络”（Network）选项卡 以显示网络视图，然后单击“的 DeviceNet”（My DeviceNet）选项卡以打开 DeviceNet 网络视图，如图 26-140 所示。所出现的视图会显示将设备添加到网络时选择的默认介质。

2. 要选择其他网络介质，并设置将其他设备添加到网络中时会使用的默认值，请单击 DeviceNet 状态栏中的“运行向导”（Run Wizard）按钮，如图 26-141 所示。

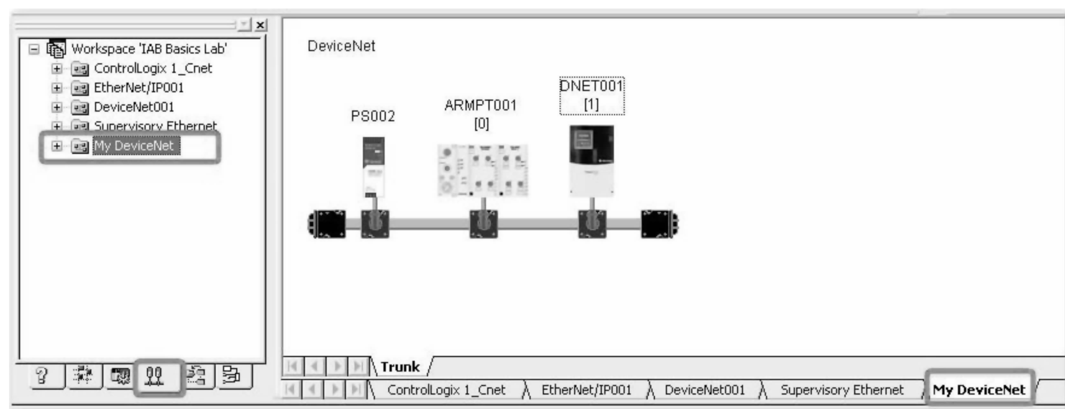


图 26-140　设备网网络图

将出现“DeviceNet 向导”（DeviceNet Wizard）对话框。可通过该向导选择 DeviceNet 物理介质的特性，如图 26-142 所示。



图 26-141 选择运行向导

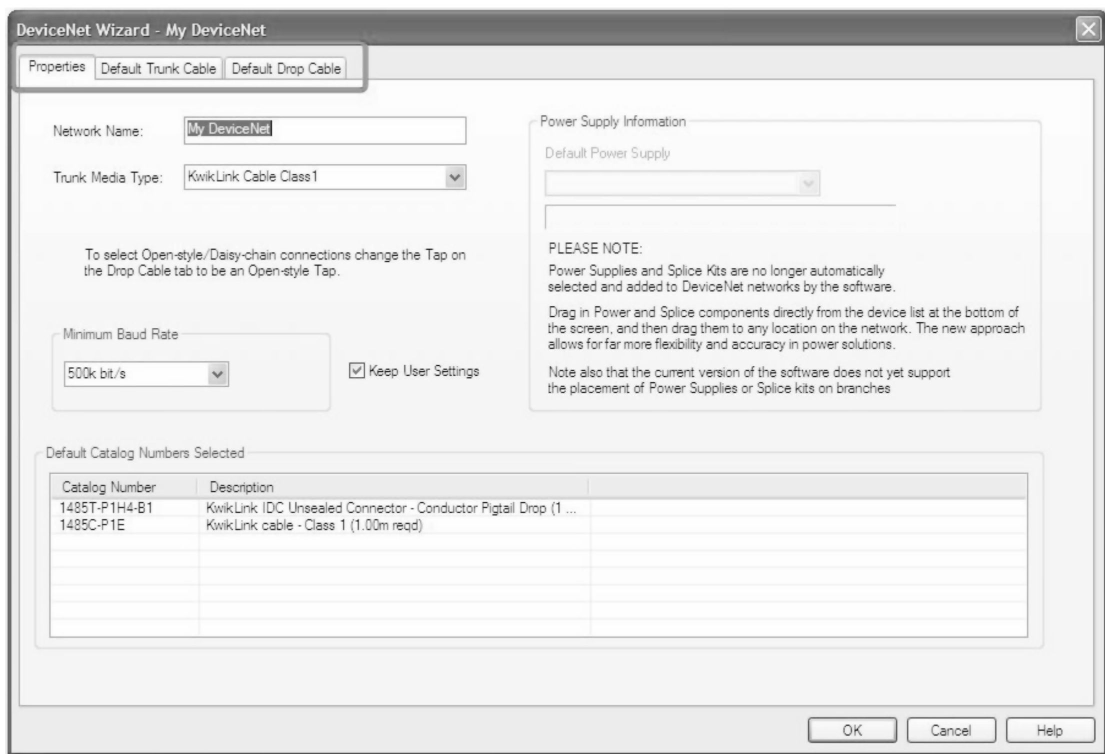


图 26-142 设备网网络向导

请注意，DeviceNet 向导有三个选项卡：

1）属性（Properties）：允许指定网络名称、干线电缆类型的默认值和最大波特率，并显示在“干线”（Trunk）和“支线电缆”（Drop Cable）选项卡中当前选定的项目。

2）默认干线电缆（Default Trunk Cable）：允许指定电缆组态（跳线、连接线或散装电缆），并选择特定的电缆和连接器。

3）默认支线电缆（Default Drop Cable）：允许指定默认分接器和支线电缆。

现在单击这三个选项卡，以查看每个选项卡中有哪些可用选项。在本基础知识实验练习中，不会详细介绍如何使用这些设置。要了解有关如何使用该向导的更多信息，请完成 DeviceNet 组态实验练习。

3. 单击“属性”（Properties）选项卡。在“干线介质类型”（Trunk Media Type）下拉列表中选择“细电缆”（Thin Cable）作为干线介质类型，然后单击“确定”（OK），如图 26-143 所示。

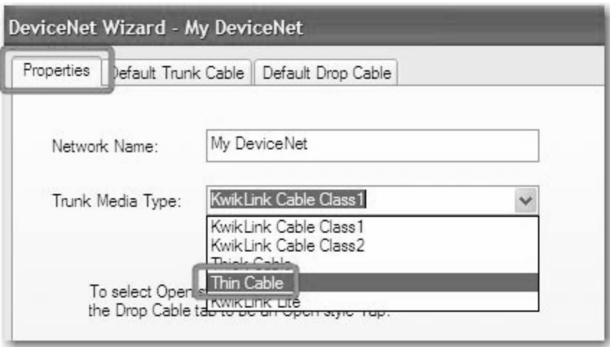


图 26-143　属性选项卡

该向导会将网络重新组态为使用细介质，并会验证网络组态。网络视图会显示组态结果。将光标移至网络元素之上，以显示有关该元素的信息，如图 26-144 所示。

如有可能，DeviceNet 向导会应用向导中指定的默认值（见图 7-133）。如果特殊的网络节点无法使用所选电缆，IAB 将应用与向导中指定的组态尽可能接近的有效选项。

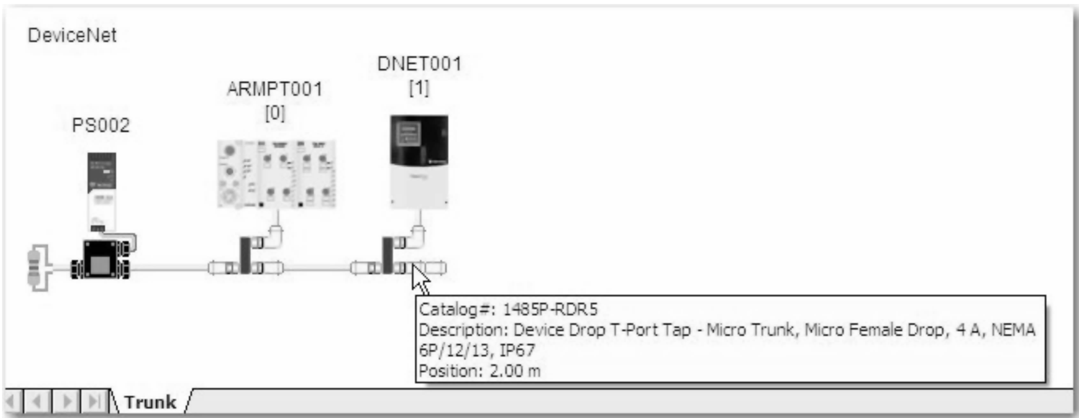


图 26-144　网络视图显示组态结果

4. 单击 DeviceNet 状态栏上的 >> 按钮，以显示有关 DeviceNet 网络的详细信息，如图 26-145、图 26-146 所示。



图 26-145　设备网状态栏

可以选中不同的框以显示关于网络视图的附加信息。单击“干线长度”（Trunk Length）和“位置”（Position）复选框，并查看信息的显示方式。同时，尝试对其他复选框进行相同的操作，如图 26-147 所示。

5. 右键单击 PowerFlex 变频器节点图标（DNET001），然后选择“通道‘.’属性”

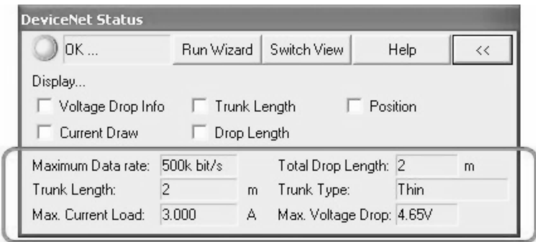


图 26-146　状态栏详细信息

（Channel‘.’Properties），如图 26-148 所示。

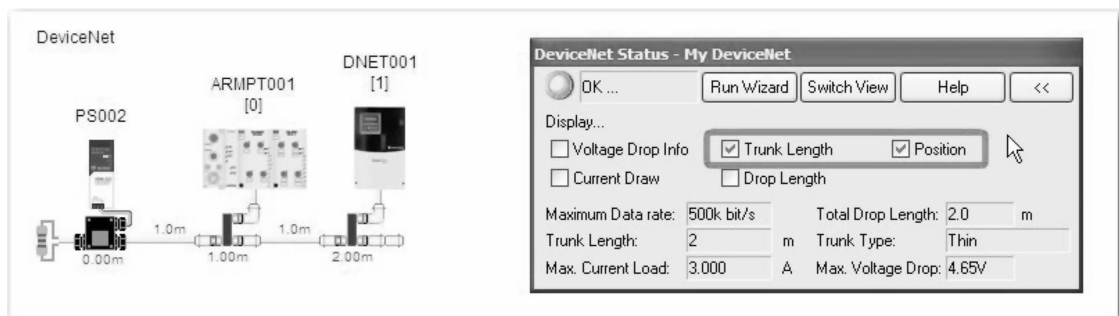


图 26-147　“干线长度”和“位置”复选框

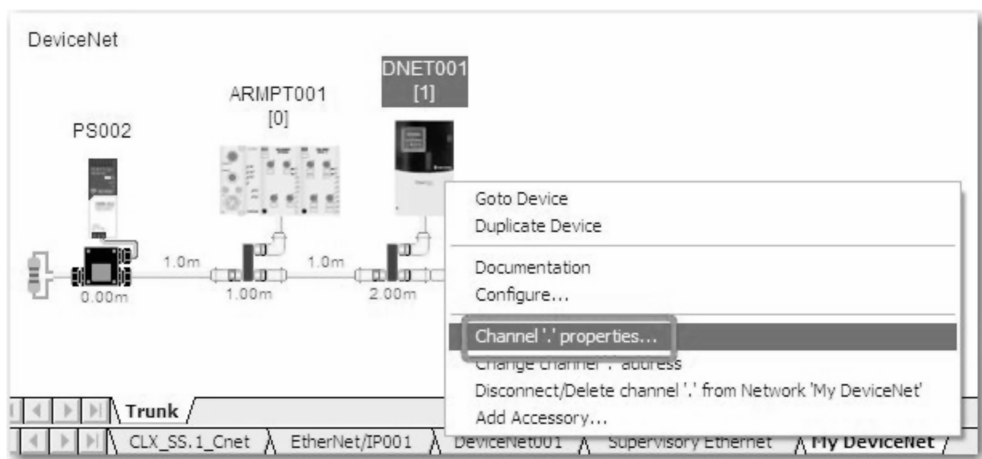


图 26-148　变频器的通道属性

将出现 DeviceNet 通道属性对话框。该对话框与 DeviceNet 向导对话框相似，只是选择的内容只能应用于特定的网络节点。在“通道属性”（Channel Properties）中进行的设置将覆盖“DeviceNet 向导”（DeviceNet Wizard）中的默认设置。请注意，该对话框的布局中也存在多个选项卡，如图 26-149 所示。

1）设备属性（Device Properties）：允许指定设备名称和地址，并显示网络上的设备位置（从网络起始处开始测量），以及在其他选项卡中选择的干线和支线电缆部件的相关信息。

2）DevicePort/DeviceBox：允许将节点连接到新建或现有 DevicePort 或 DeviceBox。

3）分接器和支线电缆（Tap and Drop Cable）：允许指定该节点所使用的分接器和支线电缆。

4）支线电缆（Trunk Cable）：允许指定将该节点连接到上一节点的支线电缆段的组态（包括长度）。

单击各选项卡以查看其中包含的信息。

请注意通道属性对话框中显示的节点的图形，如图 26-150 所示。

在 IAB 中，该图形会突出显示 DeviceNet 组态的重要方面。该图形会显示组成网络节点的元素：

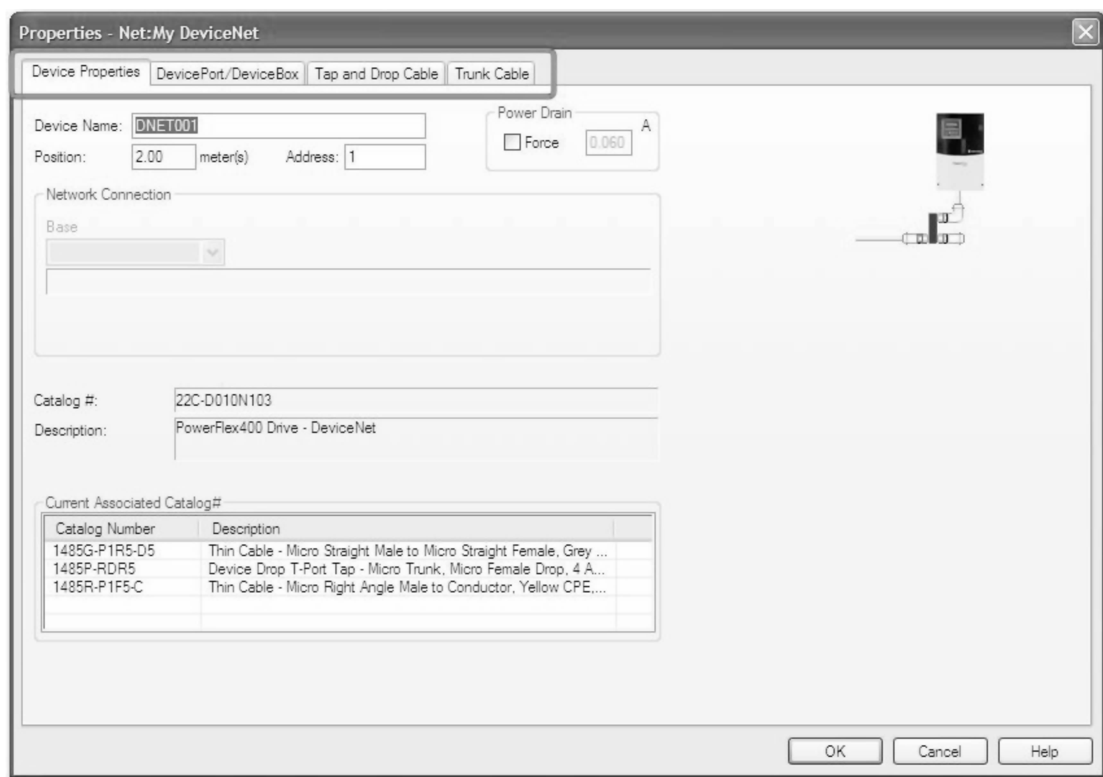


图 26-149　通道属性详细信息

1）设备；

2）支线电缆；

3）分接器；

4）将该节点连接到上一节点的电缆段。

如果移动某一节点，所有元素都会被移动（包括干线电缆段）。

在 IAB 8 及更高版本中，如果要更改 DeviceNet 节点的位置，应将该节点移动到新位置并更改干线电缆段的长度。其他节点的位置也会根据将已移动节点置于的位置及其各自的干线电缆段的长度而发生变化。可通过显示位置数据来查看节点位置，如步骤 1 所示。

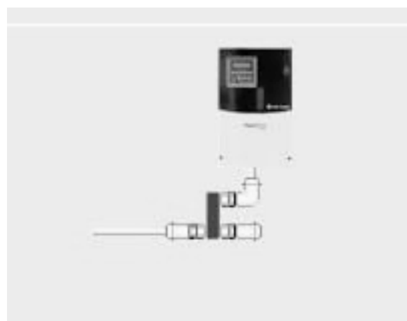


图 26-150　变频器节点图形

6. 单击“确定”（OK）接受该节点的通道属性默认值。要了解有关如何使用 DeviceNet 通道属性的更多信息，请完成 DeviceNet 组态实验练习。

7. 在 DeviceNet 视图中，将光标移至 PowerFlex 变频器（DNET 001）上方，按下左光标键，然后将该节点向左拖动，直到指示器位于电源和 ArmorPoint 节点之间，释放鼠标按键，将变频器置于其新位置，如图 26-151、图 26-152 所示。

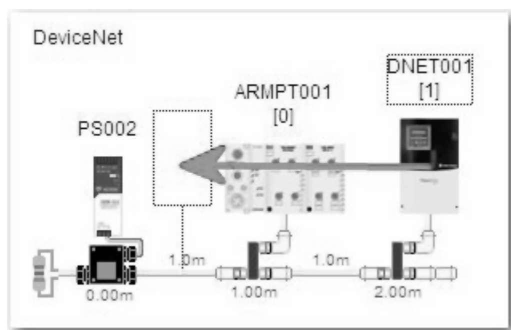


图 26-151 拖动变频器

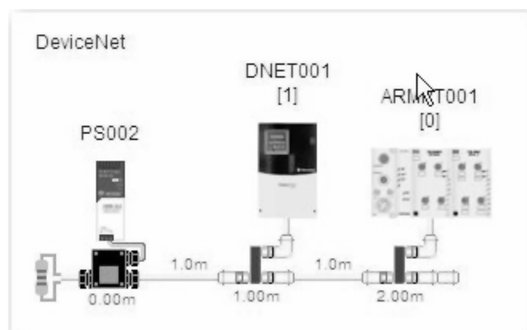


图 26-152 变频器新位置

十一、EtherNet/IP 组态特性

1. 转至“网络视图”（Network View）并单击 EtherNet/IP001 选项卡，如图 26-153 所示。

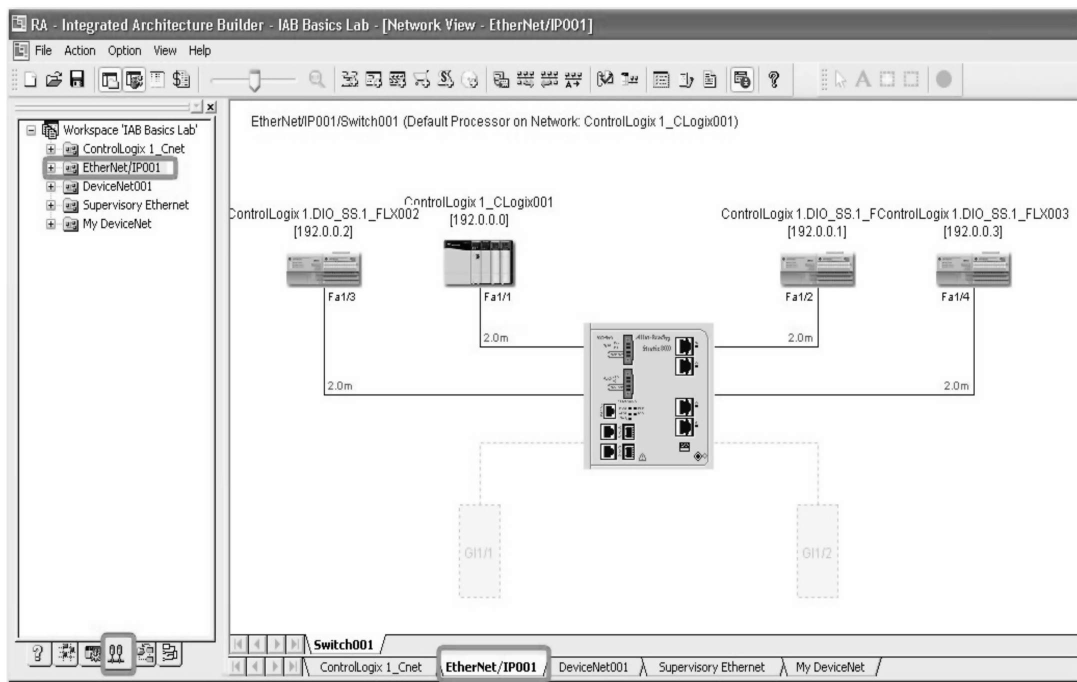


图 26-153 网络视图

2. 右键单击交换机，然后选择“交换机属性”（Switch Properties），如图 26-154 所示。IAB 会显示“交换机属性”（Switch Properties）对话框，如图 26-155 所示。

在该对话框中，可以进行以下操作：

1）指定交换机名称；

2）选择要使用的特定交换机；

3）将 SFP 光纤端口添加到交换机上（如果交换机允许使用）；

4）选择扩展模块（如果交换机允许使用）。

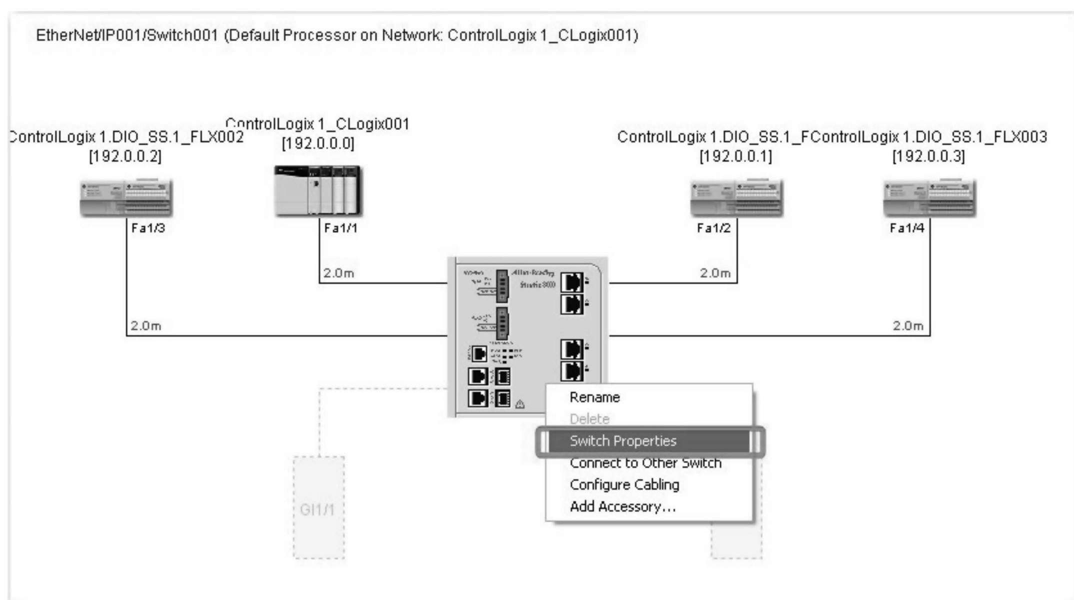


图 26-154　选择交换机属性

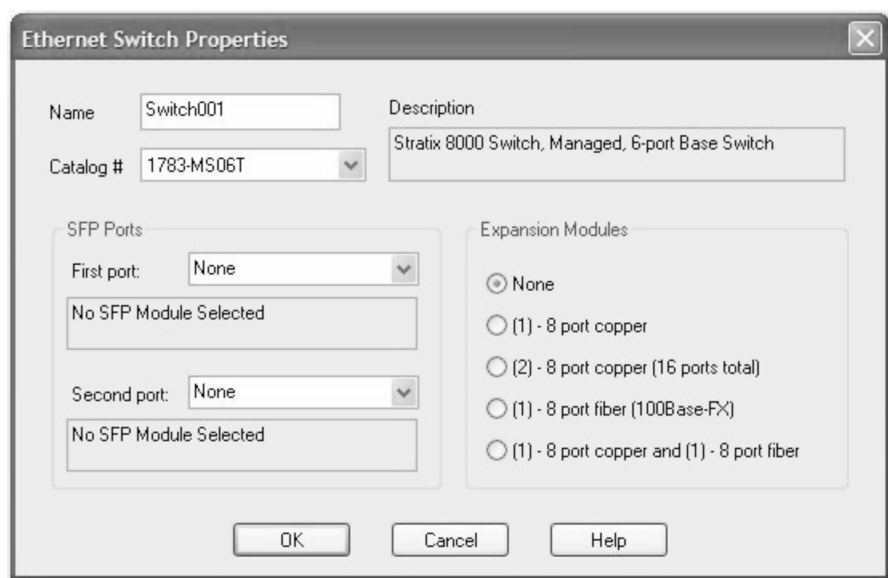


图 26-155　“交换机属性”对话框

3. 在“目录号”（Catalog #）下拉列表中，选择 1783-EMS08T 并单击“确定”（OK）。交换机会更改为所选目录号，如图 26-156 所示。

4. 右键单击交换机，然后选择“组态接线”（Configure Cabling），如图 26-157 所示。

IAB 会显示“以太网接线向导”（Ethernet Cabling Wizard），如图 26-158 所示。

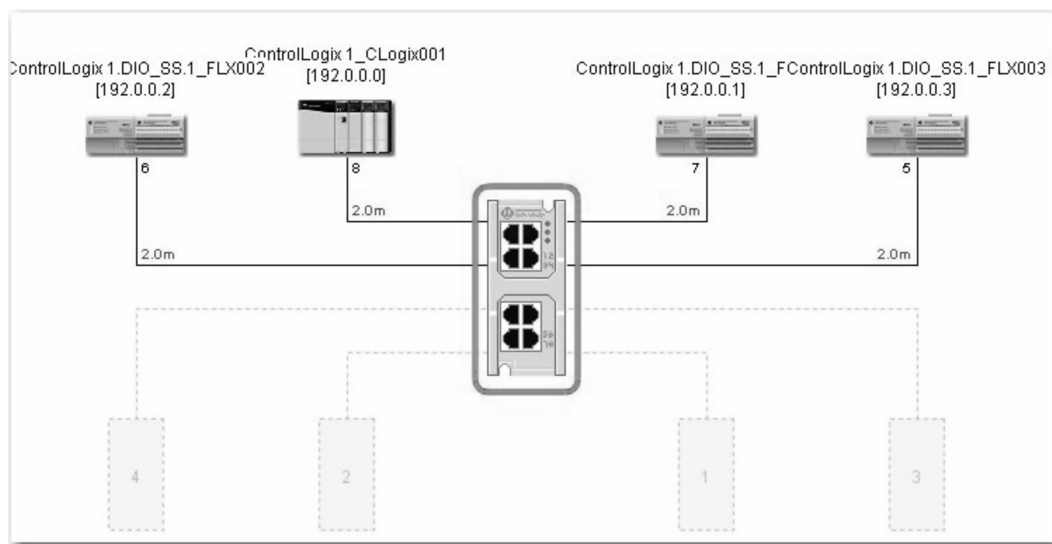


图 26-156　更改目录号

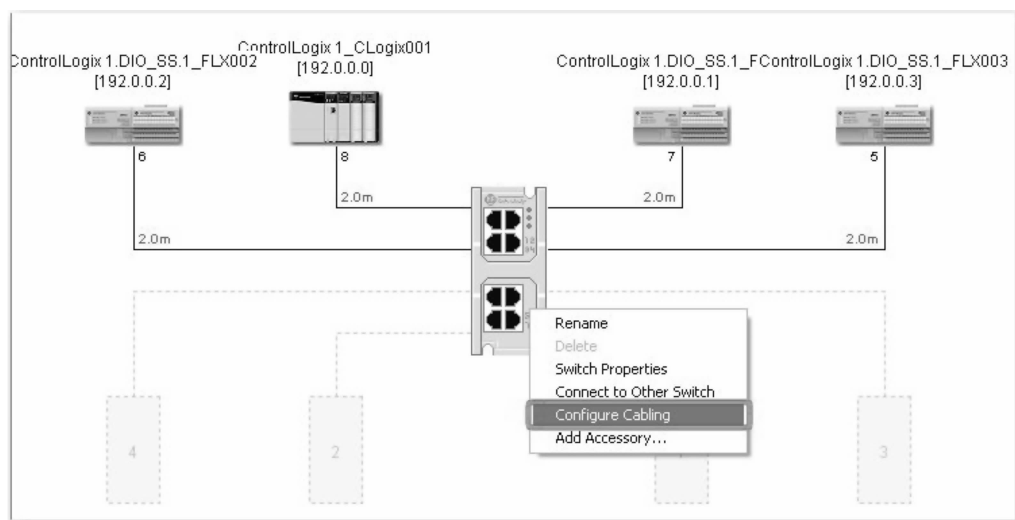


图 26-157　选择“组态接线”

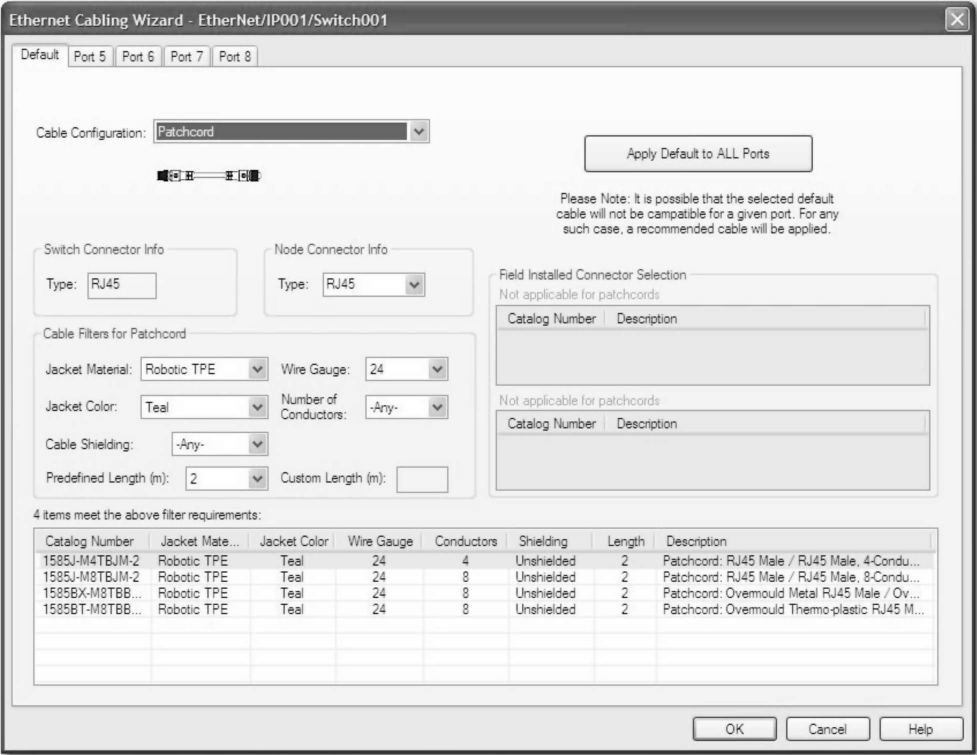


图 26-158　以太网接线向导

该对话框允许指定默认接线组态和端口特定的接线组态。可以选择：

1）电缆组态：跳线、带连接器的连接线、散装电缆和连接器等。

2）连接器类型：用于电缆两端。

3）电缆选型过滤器：护套材料、颜色、屏蔽、长度等。对话框底部的列表会显示与所选过滤器匹配的电缆。

4）现场安装连接器：如果需要单击“确定”（OK）或“取消”（Cancel）关闭对话框。

5. 从 EtherNet/IP 设备列表中，将 Stratix 6000 1783-EMS08T 交换机拖动到网络视图中（见图 26-159）。该交换机将被添加到网络中，并会自动连接到 Switch001。该交换机的一个新选项卡将被添加到网络视图中。下面介绍了如何将新交换机添加到 EtherNet/IP 网络。

6. 单击 Switch002 的选项卡。从以太网设备列表中，将 1769-L32E 处理器拖动到网络视图中，如图 26-160 所示。

IAB 会询问：是否要将这一新处理器设置为网络的默认处理器。单击“是”（Yes），如图 26-161 所示。

这些步骤仅仅显示了可用的以太网组态特性。要了解详细信息和实践内容，请参见以太网实验练习。

7. 在继续进行操作之前，请单击 IAB 工具栏上的“保存”（Save）按钮以保存之前的操作，如图 26-162 所示。

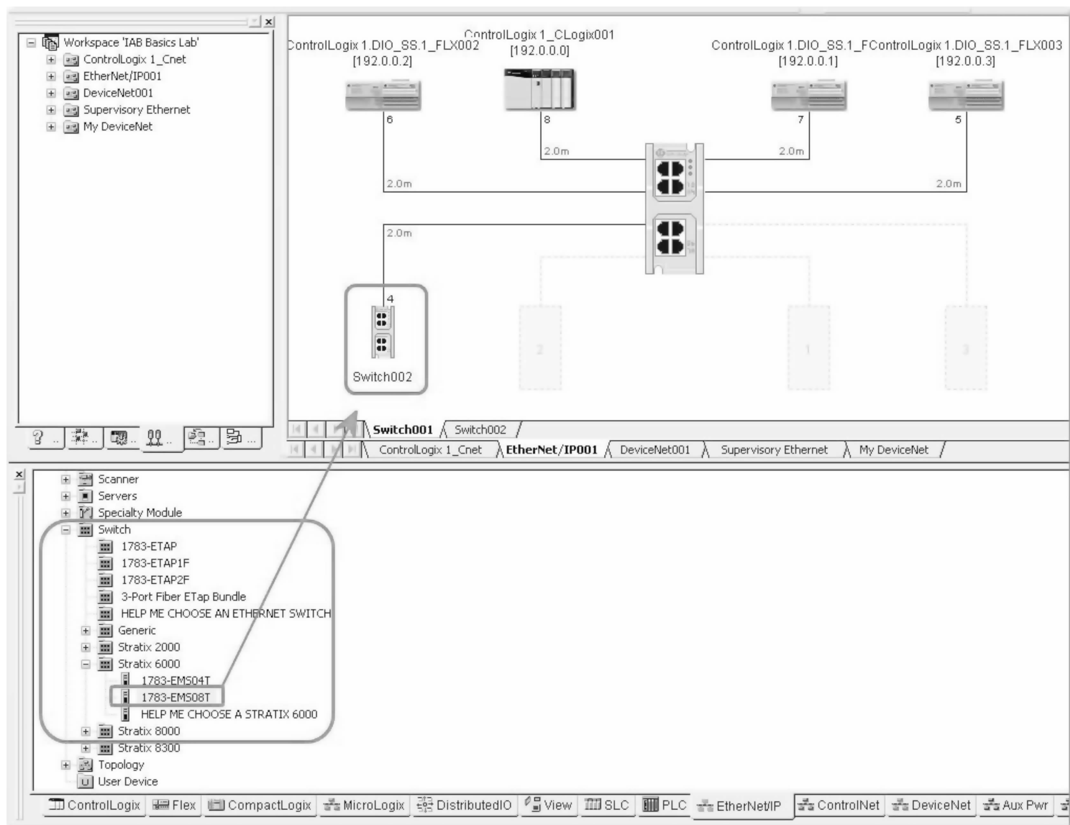


图 26-159 添加新交换机

十二、检查组态

在本练习中，将了解 IAB 提供的用于检查组态的几种功能：

1）为处理器分配 I/O 机架和设备：可以将网络上的 I/O 机架和设备与任何同样位于该网络的处理器机架相关联。这样，就可以对处理器/I/O 进行组态，从而评估处理器连接、存储器以及网络性能。

2）处理器检查：对于每个处理器机架，可以检查连接和所使用的存储器。

3）EtherNet/IP 容量检查：可以检查每个 EtherNet/IP 的 CIP 和 TCP 连接，以及“每秒信息包数”值。

4）常规检查：可以对组态进行总体检查并查看警告和错误消息。

1. 首先，采用一些设置来帮助演示 IAB 的检查功能。转至系统视图，右键单击 ControlLogix 1 子系统图标，然后选择“复制”（Duplicate），如图 26-163 所示。

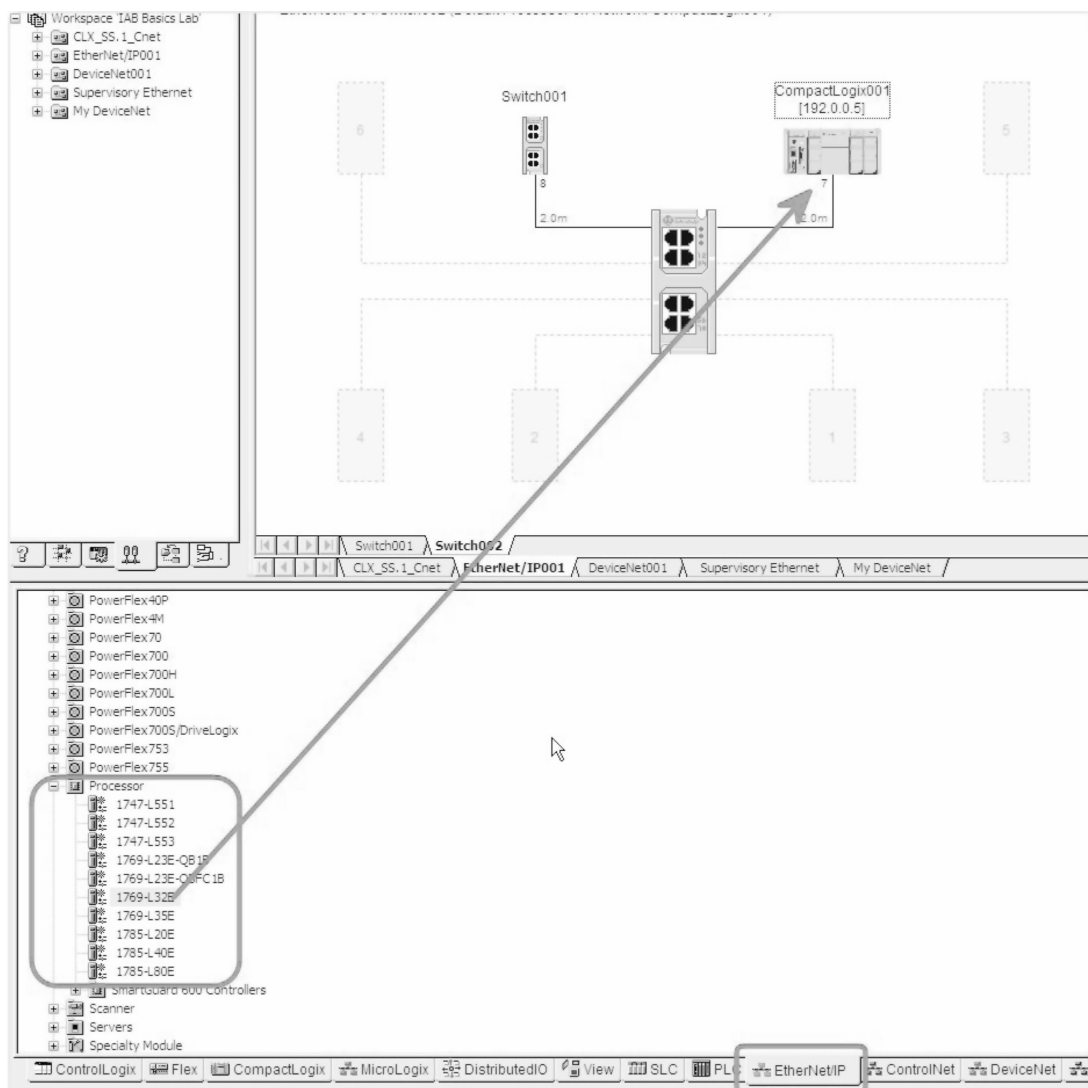


图 26-160　添加新处理器

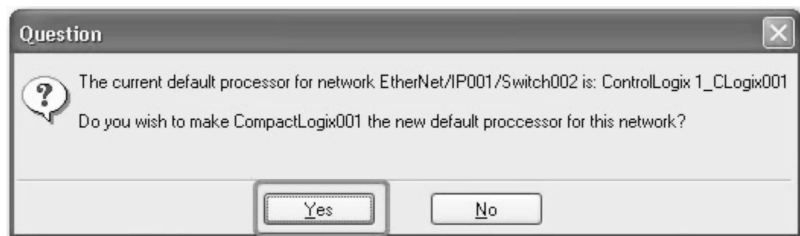


图 26-161　确认按钮



图 26-162　保存按钮

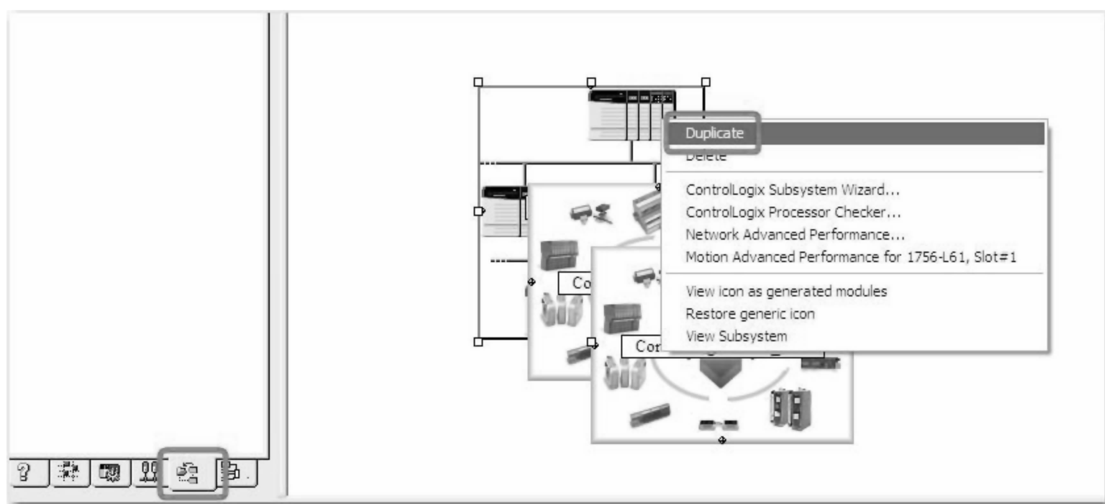


图 26-163 系统视图

IAB 将用 1min 的时间来处理该请求，然后将新的子系统图标添加到视图中。将该图标拖动到视图中方便的位置，如图 26-164 所示。

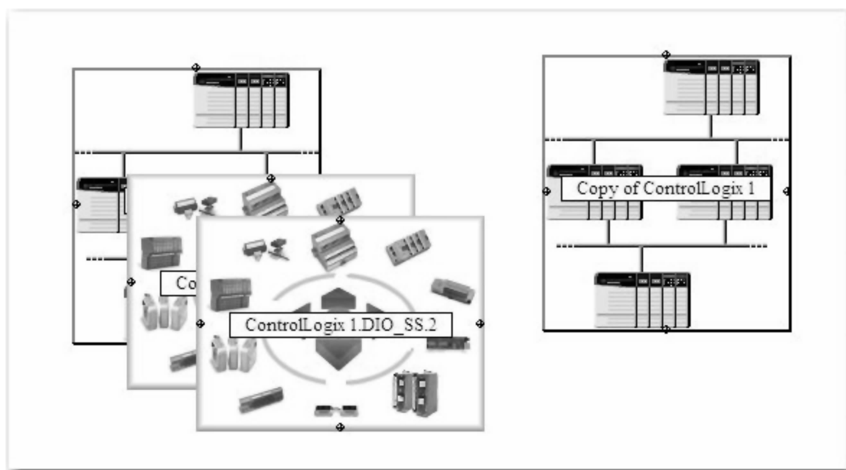
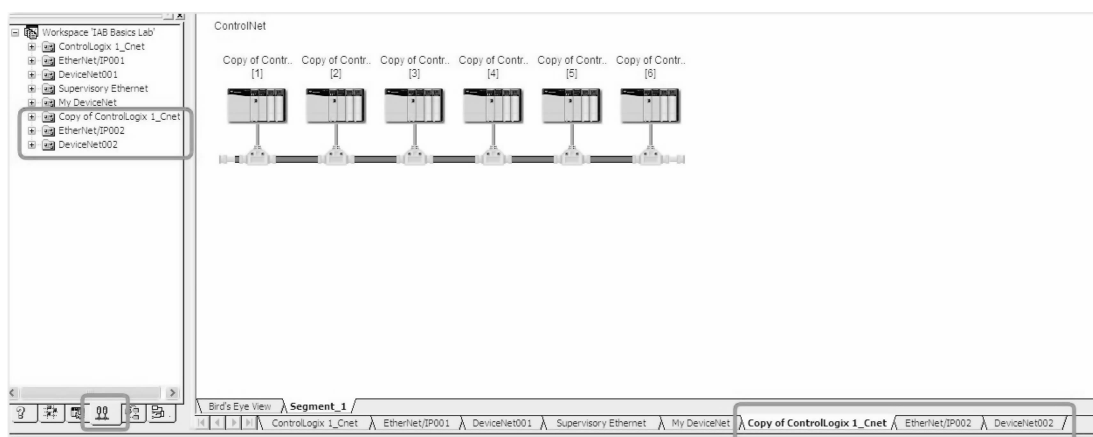
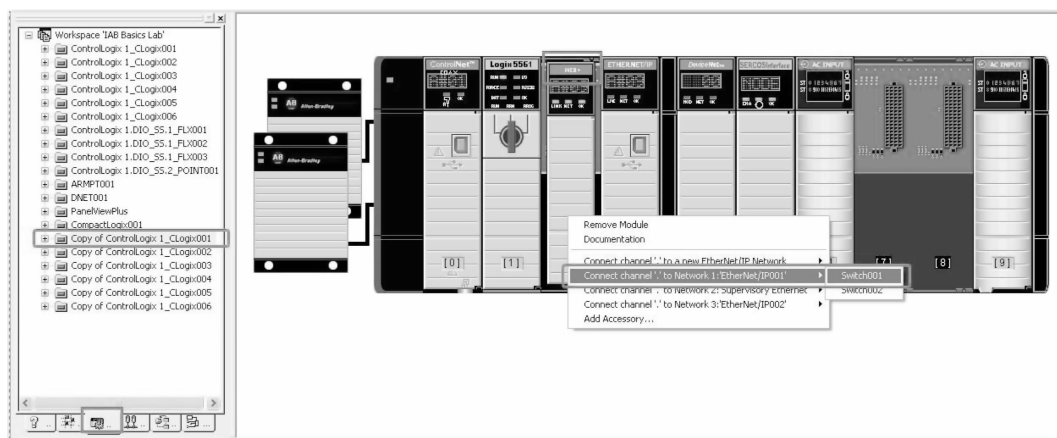


图 26-164 复制子系统

2. 转至“网络”（Network）视图。IAB 已通过复制原始 ControlLogix 1 向导中指定的三个网络创建了三个新网络。检查每个新网络的视图。Copy of ControlLogix 1_Cnet 看起来与 ControLogix 1_Cnet 相似。EtherNet/IP002 和 DeviceNet002 是空的。IAB 复制了原始 ControlLogix 1 向导中指定的所有内容（包括两个用于分布式 I/O 的附加网络），但并未将任何分布式 I/O 添加到新的附加网络，如图 26-165 所示。

3. 转至“硬件”（Hardware）树，然后选择 Copy of ControlLogix 1_CLogix001。它是复制版的 ControlLogix 子系统中的处理器机架。右键单击 1756-EWEB 模块，然后选择“将通道‘.’连接到网络 1：‘EtherNet/IP001’ >交换机 1”（Connect channel ‘.’ To Network 1：’EtherNet/IP001’ >Switch 1），如图 26-166 所示。


图 26-165　网络视图


图 26-166　硬件视图

IAB 会询问：是否要将该处理器机架（Copy of ControlLogix 1）设置为 EtherNetIP/001 的默认处理器机架。单击“是”（Yes），如图 26-167 所示。



图 26-167　确认按钮

现在，EtherNet/IP001 的网络视图会显示两个已连接的 ControlLogix 机架，如图 26-168 所示。

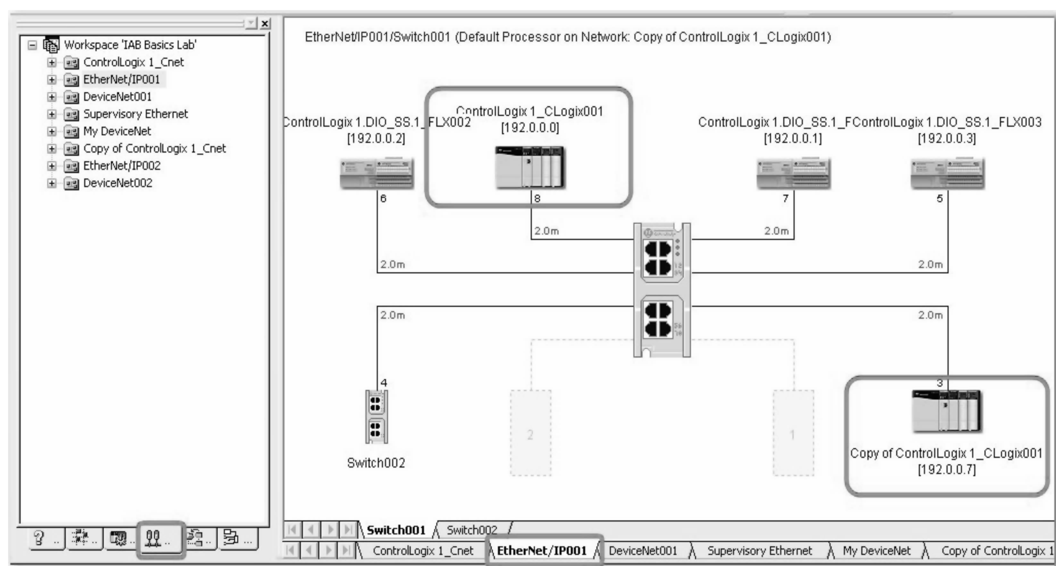


图 26-168 EtherNet/IP001 网络视图

4. 打开 EtherNet/IP002 的网络视图，将 1794-AENT 适配器从设备列表拖动到网络中，如图 26-169 所示。

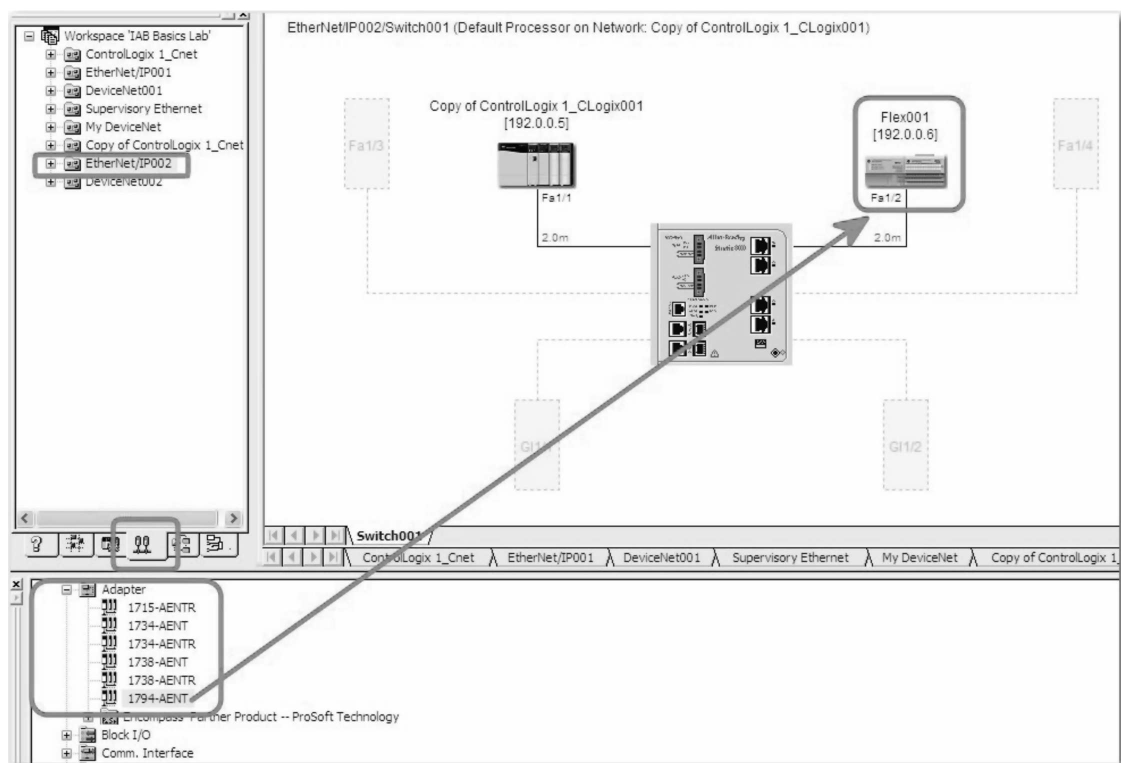


图 26-169 EtherNet/IP002 网络视图

5. 在 EtherNet/IP002 视图中双击 Flex001 节点。IAB 会打开机架视图。右键单击机架，然后选择“组态机架”（Configure Chassis），如图 26-170 所示。

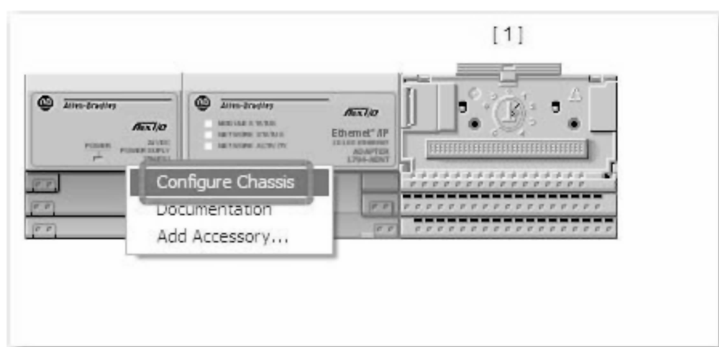


图 26-170 机架组态

6. 在“组态机架”（Configure Chassis）对话框中，将插槽数设置为 8，然后单击“确定”（OK），如图 26-171 所示。IAB 会将插槽添加到机架中，如图 26-172 所示。

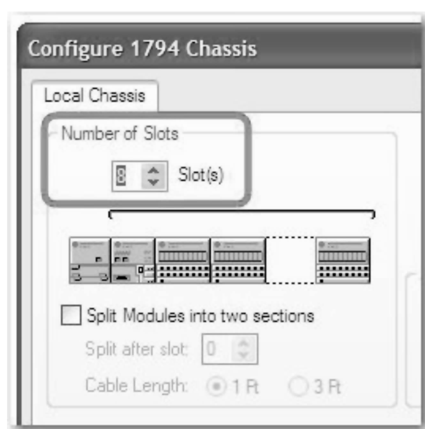


图 26-171 机架组态

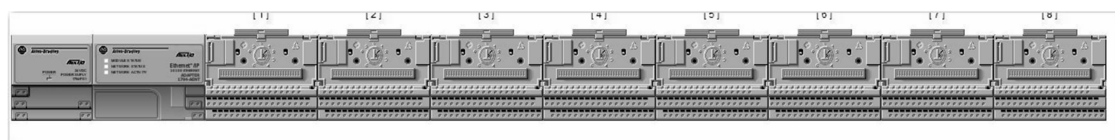


图 26-172 八个插槽

7. 将八个 1794-IJ2I 专用模块拖动到机架中，如图 26-173 所示。

十三、为处理器机架分配 I/O 和设备

1. 单击 IAB 工具栏上的“分配处理器连接”（Assign Processor Connections）按钮。IAB 会显示“为处理器分配远程机架/设备”（Assign Remote Chassis/ Device to Processor）对话框，如图 26-174 所示。

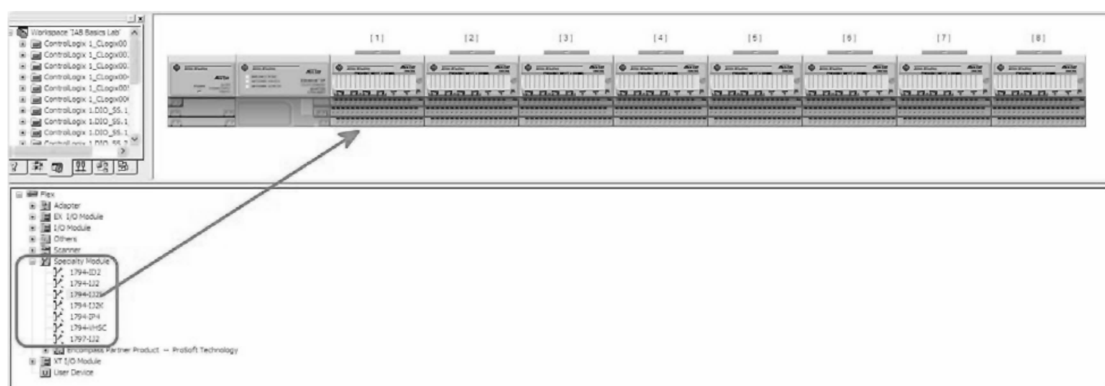


图 26-173 拖动模块

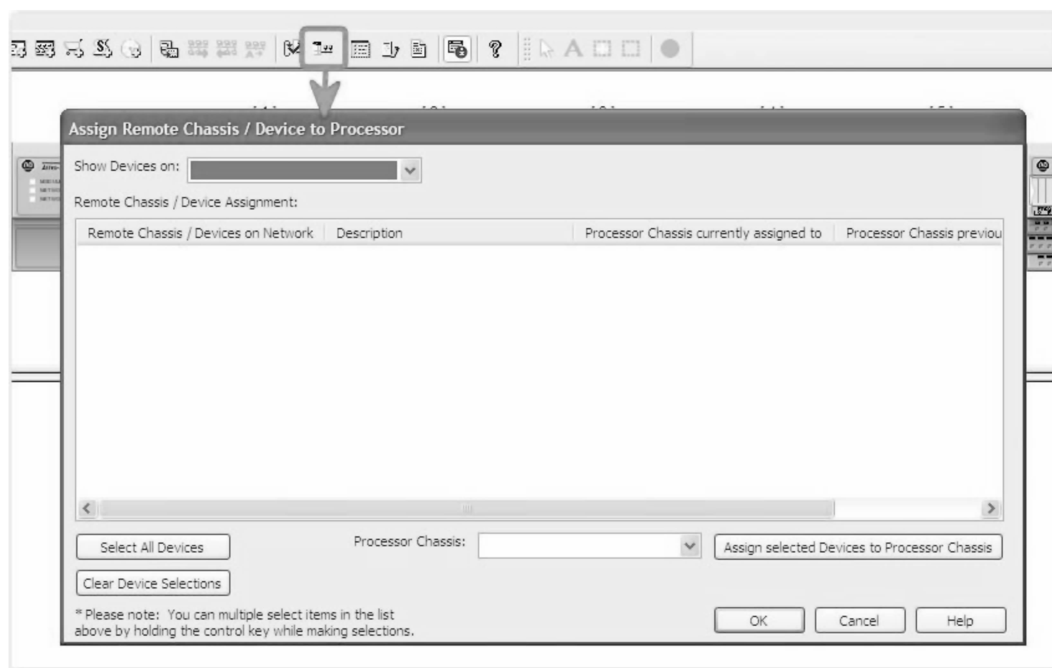


图 26-174 远程机架/设备分配视图

2. 单击“显示位于以下位置的设备:”（Show Devices on:）下拉列表。该下拉列表会显示，已在组态中创建的网络。选择 ControlLogix 1_Cnet，如图 26-175 所示。

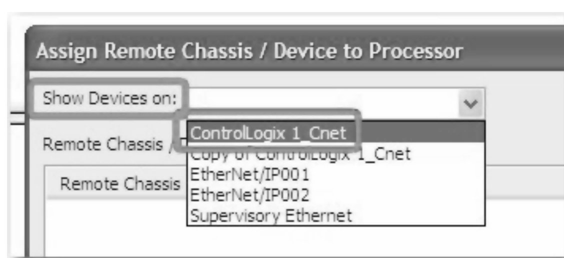


图 26-175 设备显示视图

IAB 会列出已连接到所选网络的所有设备和远程 I/O 机架。在该例中，将列出 ControlLogix 子系统的五个远程机架（处理器机架未被列出），如图 26-176 所示。

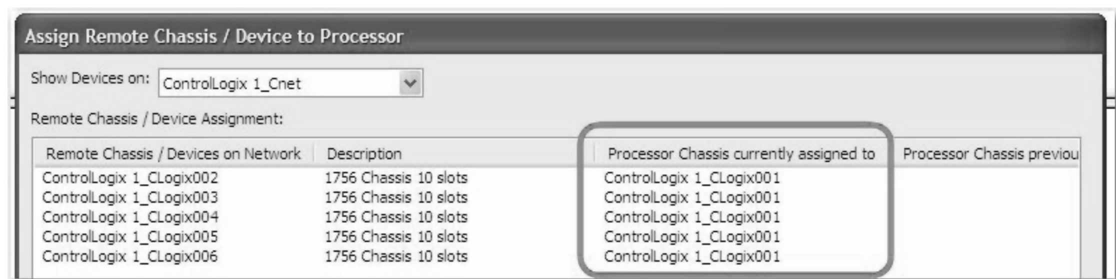


图 26-176　远程机架

第三列显示的是目前分配给每个远程 I/O 机架的处理器。在该例中，所有机架都会被自动分配给由向导创建的处理器。

3. 在“显示位于以下位置的设备:”（Show Devices on:）下拉列表中选择 EtherNet/IP001，如图 26-177 所示。现在，将列出先前通过 DIO 向导创建的三个 FLEX I/O 机架。请注意，已分配的处理器当前是 ControlLogix 1 子系统中的处理器。这也是在使用 DIO 向导创建 I/O 时设定的。

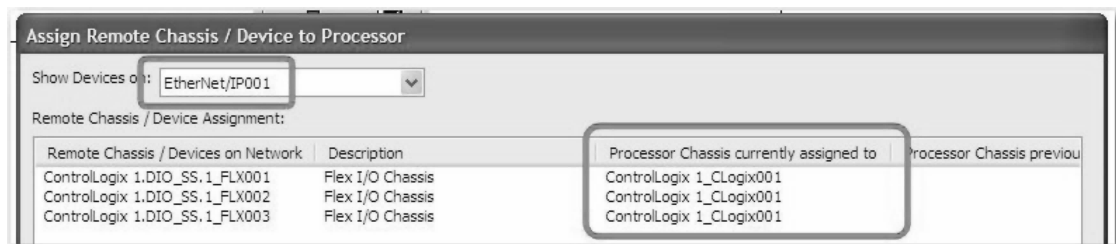


图 26-177　设备显示视图

4. 假设要使这两个 FLEX I/O 机架与 Copy of ControlLogix 子系统中的 ControlLogix 处理器相关联。在设备列表中，单击 ControlLogix 1. DIO_SS. 1_FLX001，然后按住 Ctrl 键并单击 ControlLogix 1. DIO_SS. 1_FLX003。现在，两台设备已被选定，如图 26-178 所示。

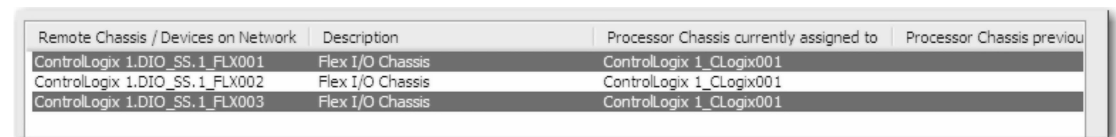


图 26-178　设备列表图

在“处理器机架:”（Processor Chassis:）下拉列表中，请注意，由于 ControlLogix 1 和 Copy of ControlLogix 1 这两个处理器都连接到 EthetNet/IP001，因此会将其列出。选择 Copy of ControlLogix 1_CLogix001，然后单击“为处理器机架分配所选设备”（Assign selected Devices to Processor Chassis）按钮，如图 26-179 所示。

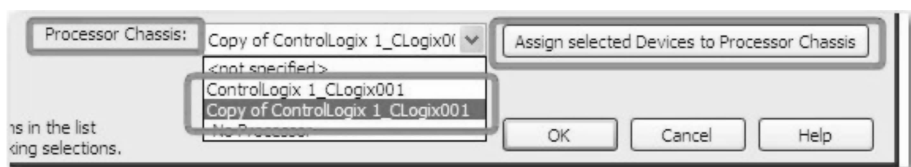


图 26-179 处理器机架视图

机架和设备列表现在将显示已分配给 Copy of ControlLogix 1 处理器的两个所选机架，如图 26-180 所示。

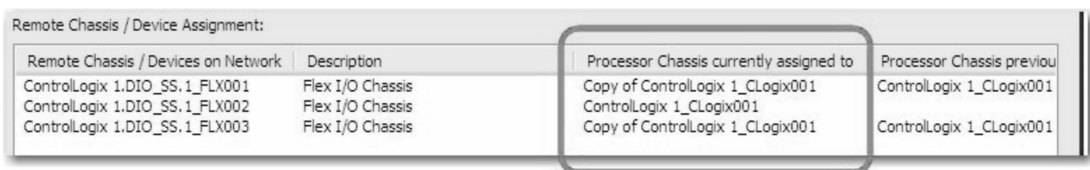


图 26-180 处理器机架分配视图

单击“确定”（OK）保存更改，然后关闭对话框。

不仅可以为处理器分配 I/O 机架，还可以为其分配其他设备，如变频器和 HMI。需要牢记的一条准则是，只能将机架和设备分配给处于同一网络上的处理器。例如，如果在 IAB 中创建两个 EtherNet/IP 网络，则即使将网络交换机互相连接，也无法将处于其中一个网络上的处理器分配给另一个网络上的设备。通常，创建一个具有多台交换机的 EtherNet/IP 网络比为每台交换机创建单独的网络要好。

5. 转至 Supervisory Ethernet 网络视图，右键单击 PanelView Plus 终端，然后选择“通道‘.’属性”（Channel‘.’Properties）。将出现“通道属性”（Channel Properties）对话框，如图 26-181 所示。

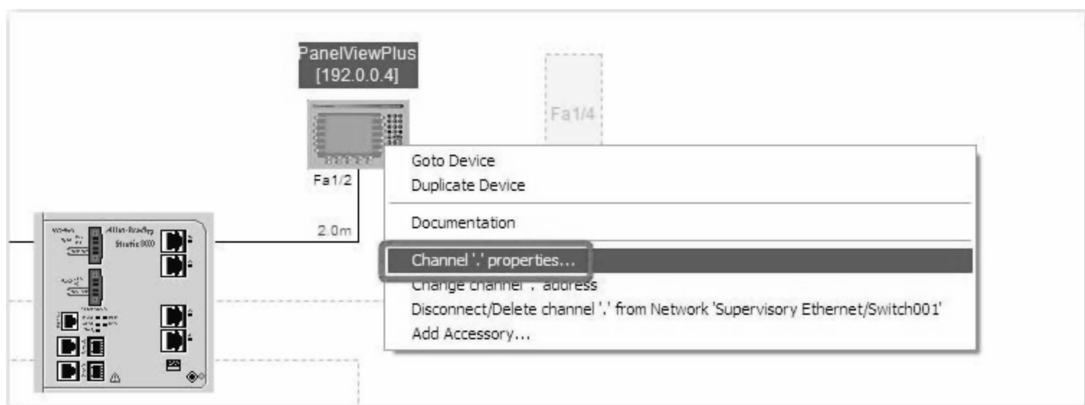


图 26-181 以太网管理网络视图

请注意，可以从“通道属性”（Channel Properties）对话框中设置关联的处理器机架，并且“通道属性”（Channel Properties）对话框可能包含其他与性能相关的设置，例如 HMI 属性。单击

两个对话框中的“取消”（Cancel）以继续，如图 26-182 所示。

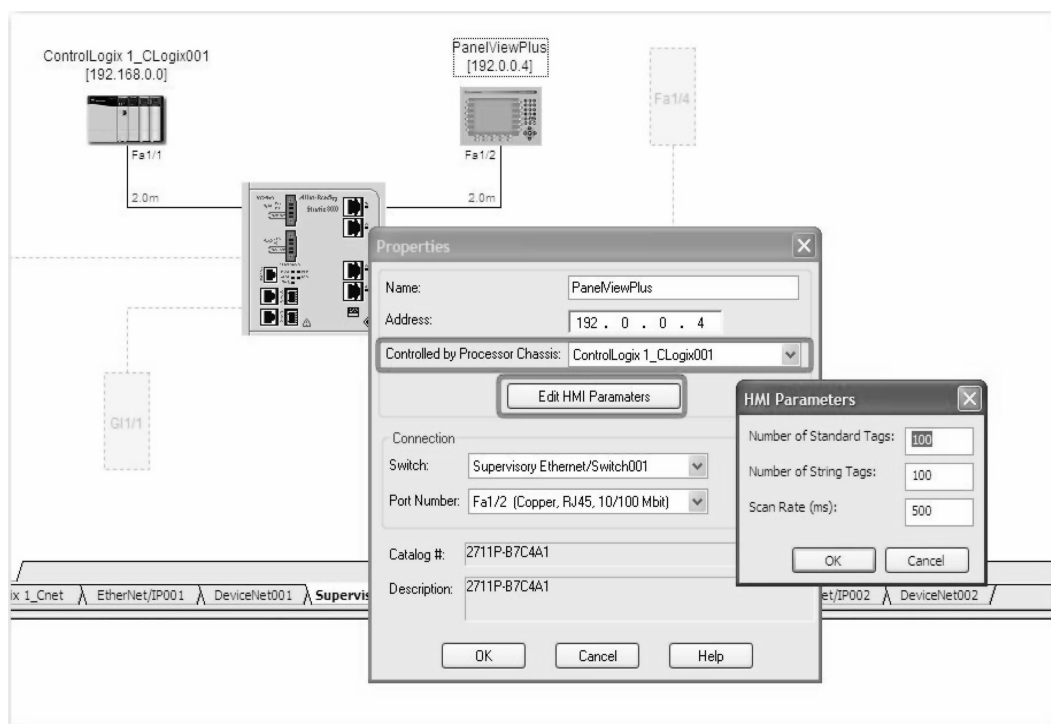


图 26-182　面板的通道属性

十四、运行处理器检测器

转至 ControlLogix 1_CLogix001 的机架视图，右键单击处理器模块，然后选择“运行处理器检测器…”（Run Processor Checker…），如图 26-183 所示。

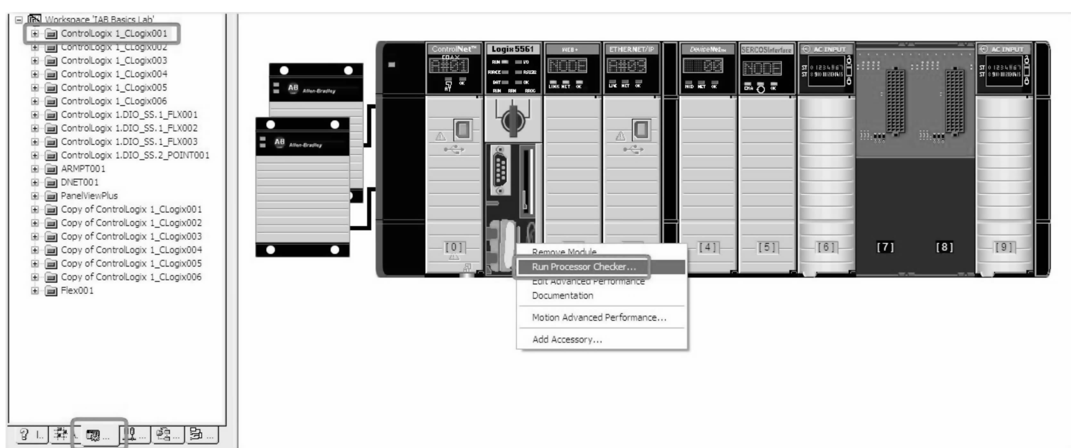


图 26-183　机架视图

将出现子系统检测器显示界面，如图 26-184 所示。

处理器、检测器、对话框会显示以下内容：

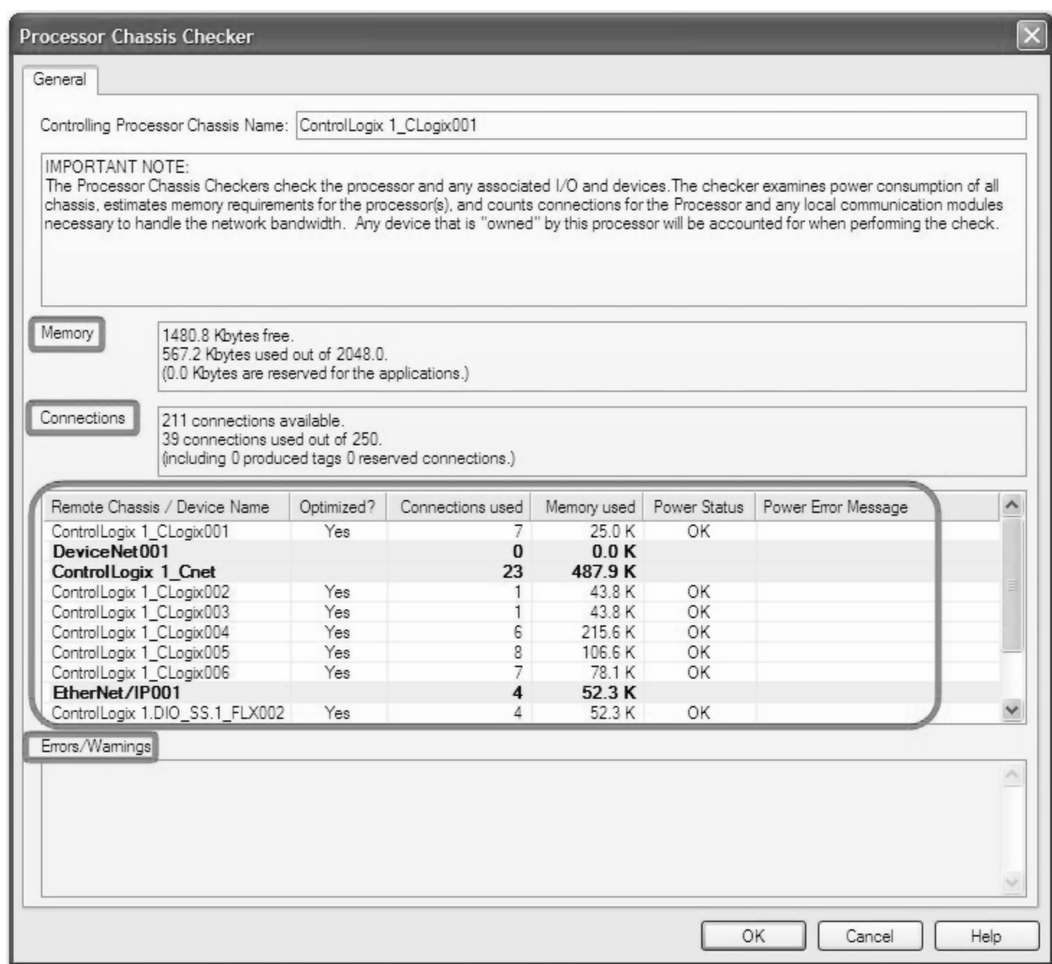


图 26-184　子系统检测器显示图

1）内存（Memory）：所需的和可用的处理器内存（根据经验法则）。

2）连接（Connections）：所需的和可用的总控制器连接。

3）连接信息（Connection information）：用于列出与处理器关联的每个网络和设备/机架、连接是直接连接还是机架优化连接、连接和所用内存，以及电源状态（对于远程 I/O 机架为背板电源；不会提供任何现场电源计算）。

4）警告和错误（Warnings and Errors）：警告是可能想要考虑更改的项目，但不是有效组态所必需的项目。错误是为了使组态有效而必须修正的项目。

对于该处理器来说，所有组态看起来都非常正常。单击“确定”（OK）关闭对话框。

十五、检查以太网容量

转至 EtherNet/IP001 的网络视图，然后单击“星型以太网拓扑”（Star Ethernet Topology）对话框中的“容量”（Capacity）按钮，如图 26-185 所示。

IAB 会显示“EtherNet/IP 容量”（EtherNet/IP Capacity）对话框，如图 26-186 所示。

请注意，该对话框包含用于已连接到网络的每个处理器的选项卡。对于每个处理器，对话框会显示：

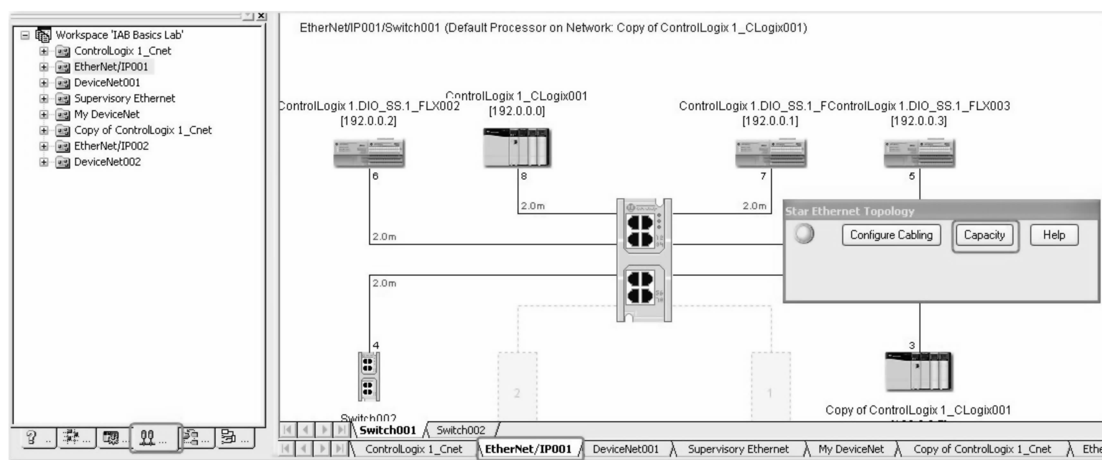


图 26-185　EtherNet/IP001 网络视图

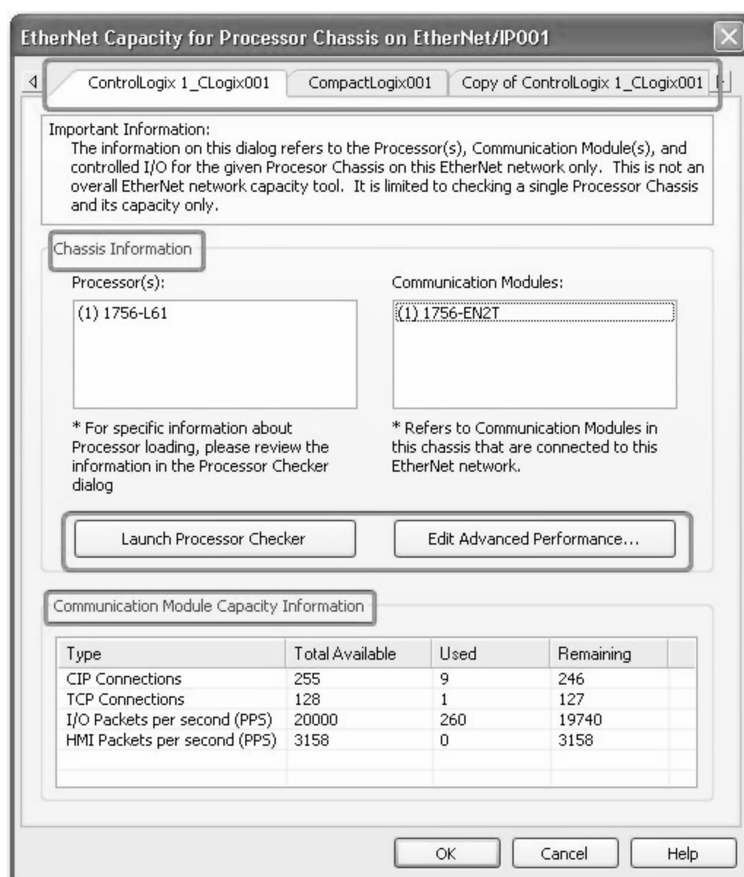


图 26-186　“EtherNet/IP 容量”对话框

1）机架信息（Chassis Information）：已连接到该网络的处理器和通信模块。

2）通信模块容量信息（Communication Module Capacity Information）：可用、已用和剩余的 CIP 连接、TCP 连接、每秒 I/O 数据包数和每秒 HMI 数据包数。

还要注意的是，可以从“以太网容量”（Ethernet Capacity）对话框访问“处理器检测器和高级网络性能”（Processor Checker and Advanced Network Performance）对话框。请参见性能检测实验练习，以了解关于 IAB 中的以上各项和性能检测的更多信息。

单击“确定”（OK）关闭“以太网容量”（Ethernet Capacity）对话框。

十六、进行常规检查

1. 单击 IAB 工具栏中的“常规检查”（General Check）按钮。IAB 会显示该项目的“常规检查”（General Check）对话框，如图 26-187 所示。

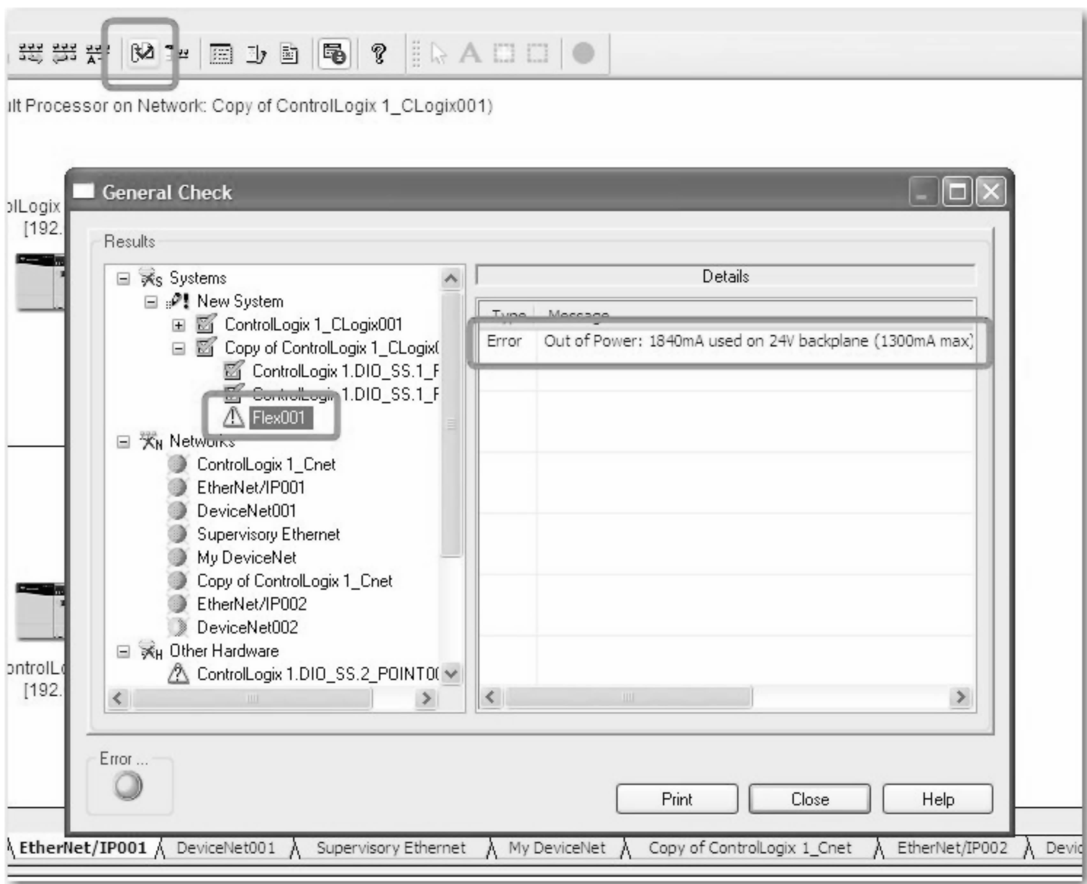


图 26-187 “常规检查”对话框

“常规检查”（General Check）对话框会列出 IAB 组态中的系统和子系统、网络及其他硬件。绿色指示灯表明检查结果正常，红色三角形表明存在错误或警告。错误和警告消息会显示在对话框的右侧。在该项目中，存在一个与 FLEX I/O 机架相关的错误。错误消息指示机架“断电”。这表示该机架中的电源无法为机架中的模块提供足够大的背板电流。

2. 要修正该错误，请转至“硬件”（Hardware）选项卡并选择 Flex001 项，以显示机架视图，如图 26-188 所示。

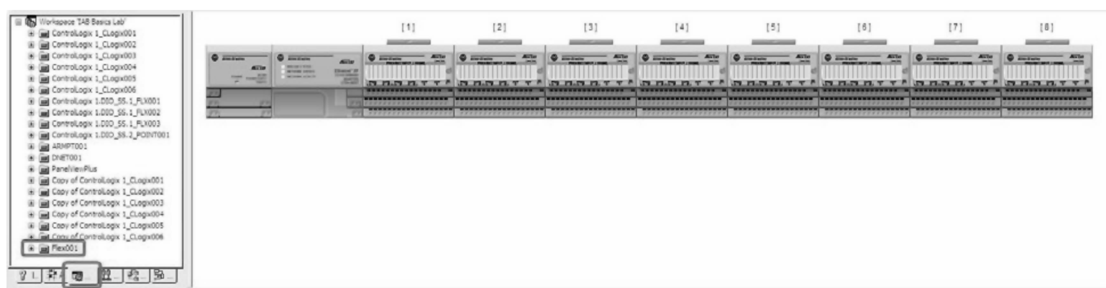


图 26-188　“硬件”选项卡

3. 右键单击适配器或电源，然后选择“组态机架”（Configure Chassis），如图 26-189 所示。

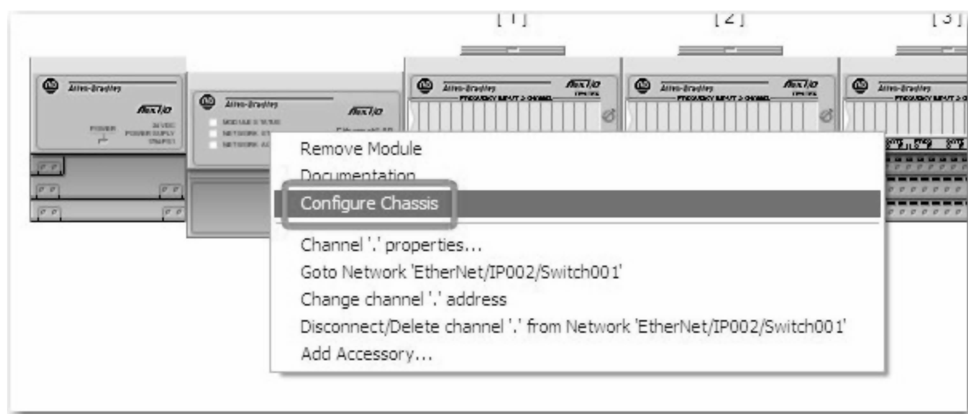


图 26-189　选择“组态机架”

“组态机架”（Configure Chassis）对话框也会显示该错误，如图 26-190 所示。

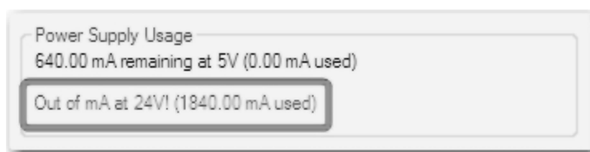


图 26-190　显示电源错误

4. 在“电源”（Power Supply）区域中，选择 1794-PS3 电源。错误指示便会消失。所选电源可以满足该机架中模块的背板要求，如图 26-191 所示。

单击“确定”（OK）。

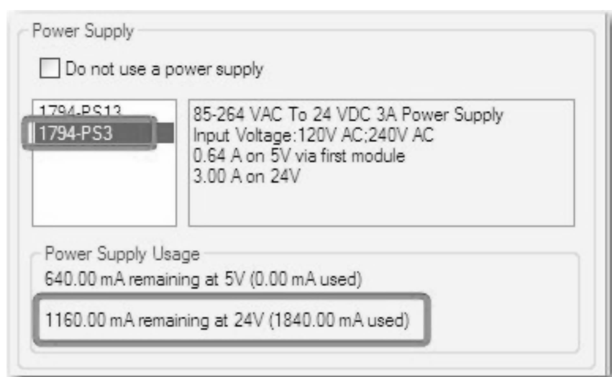


图 26-191　选择电源模块

5. 重新打开“常规检测器”（General Checker）。仍存在两个红色三角形。这两个红色三角形均显示警告消息，指示可能需要在完成组态前考虑的内容，但并不是必需的。在本例中，两条警告消息均指示组态中的项目尚未连接到辅助电源网络。IAB 允许将设备连接到辅助电源“网络”并选

择配电物理介质，但这并不是针对任何组态的要求。忽略这些警告，然后单击“关闭”（Close），如图 26-192 所示。

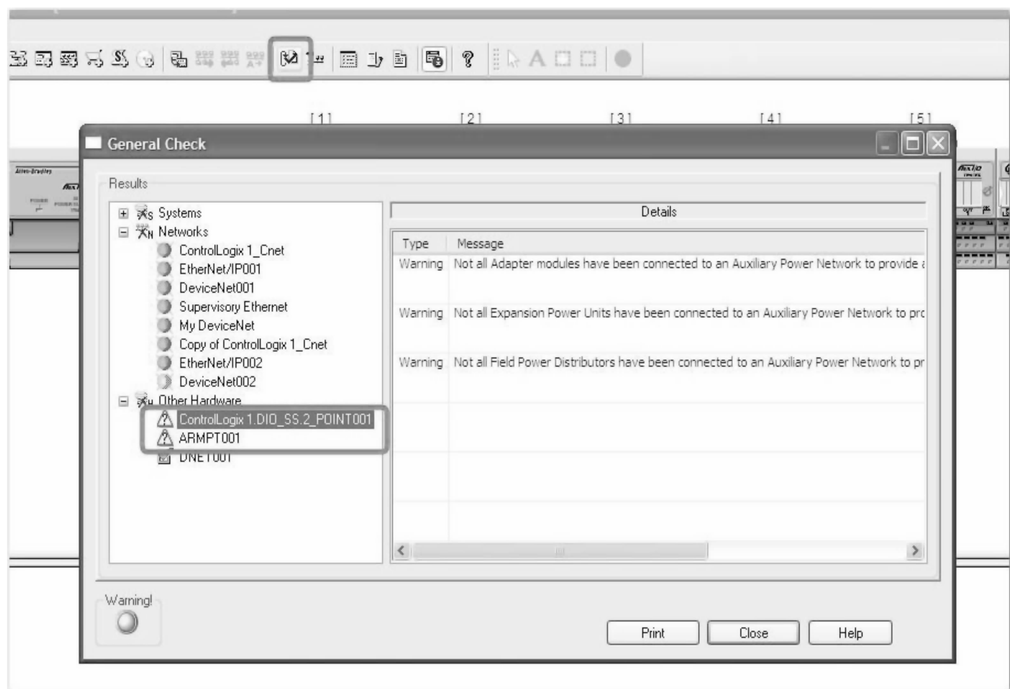


图 26-192 “常规检测器”对话框

6. 至此，本实验的组态检查部分结束。单击“保存”（Save）按钮保存操作（见图 26-193）。接下来，将了解架构视图。

十七、使用架构视图

组态设备和网络时，IAB 会自动构建所有已创建内容的架构视图。可以使用架构视图来获取已创建架构的总体视图，并且可以看到网络互连的情况。本练习将介绍如何使用架构视图。



图 26-193 “保存”按钮

1. 要显示架构视图，请单击“架构”（Architecture）选项卡。使用缩放滑块，使整个架构视图显示在窗口中，如图 26-194 所示。

架构视图会显示所有网络和已连接到网络的所有设备。已创建但并未连接到网络的设备也会显示在视图上方。

2. 按下 Ctrl 键和 Shift 键，然后在架构视图中右键单击，以重画视图，如图 26-195 所示。

使用 Ctrl-Shift > 右键单击操作时要格外小心，因为这将撤销之前进行的任何重新排列视图的操作。如果是第一次查看架构视图，这是一个很好的方法，可以确保显示所有连接。

3. 使用缩放滑块以放大视图并显示更多详细信息。请注意设备和网络名称的显示方式，如图 26-196 所示。

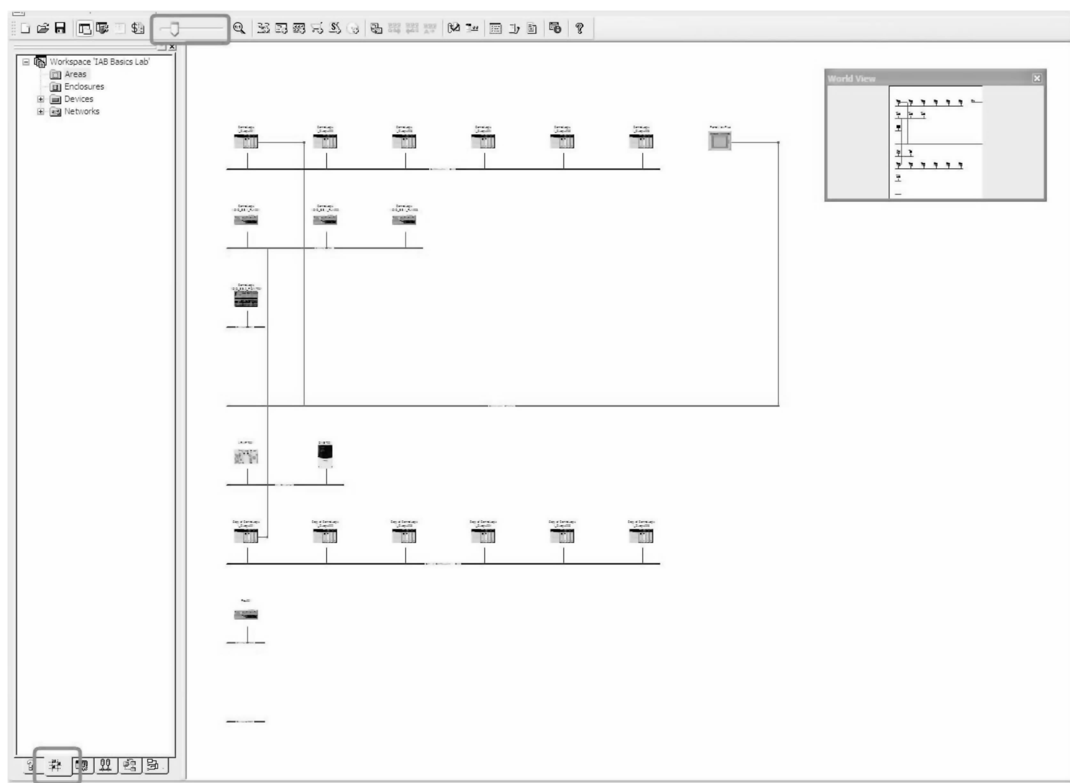


图 26-194　架构视图

4. 在“全局视图”（World View）框中，将该视图框四处拖动，以显示该视图的不同部分，如图 26-197 所示。

5. 在架构视图中，可以四处移动设备，以符合演示的要求。假设，希望 PanelView Plus 终端直接出现在视图上方线路上。将光标移至终端上，按住鼠标左键，将变频器拖动到新位置。拖动时，要注意，变频器图像有时会显示为较浅颜色或“灰显”，有时会显示为较深颜色。只能将设备放在图像显示为深颜色且未灰显的位置。直接将变频器拖动到上方线路中最后一个 ControlLogix 机架的右侧，然后释放鼠标按键，如图 26-198 所示。

IAB 现在会显示位于新位置的终端，还会重画网络连接，如图 26-199 所示。

6. 也可拖动设备组。拖动选中的整个第二行的 ControlLogix 机架，如图 26-200 所示。

将光标置于选择框内，按下鼠标左键。所选设备将出现在用交叉的平行线画出阴影的红框中，如图 26-201 所示。

按住鼠标左键，并将这些设备拖动到 PanelView Plus 终端过去所在的行。设备处于可以放置所有设备的位置之后，该框会从红色的交叉平行线变为绿色，并会打开，如图 26-202 所示。

释放鼠标按键。IAB 会重画设备位于新位置的架构视图，如图 26-203 所示。

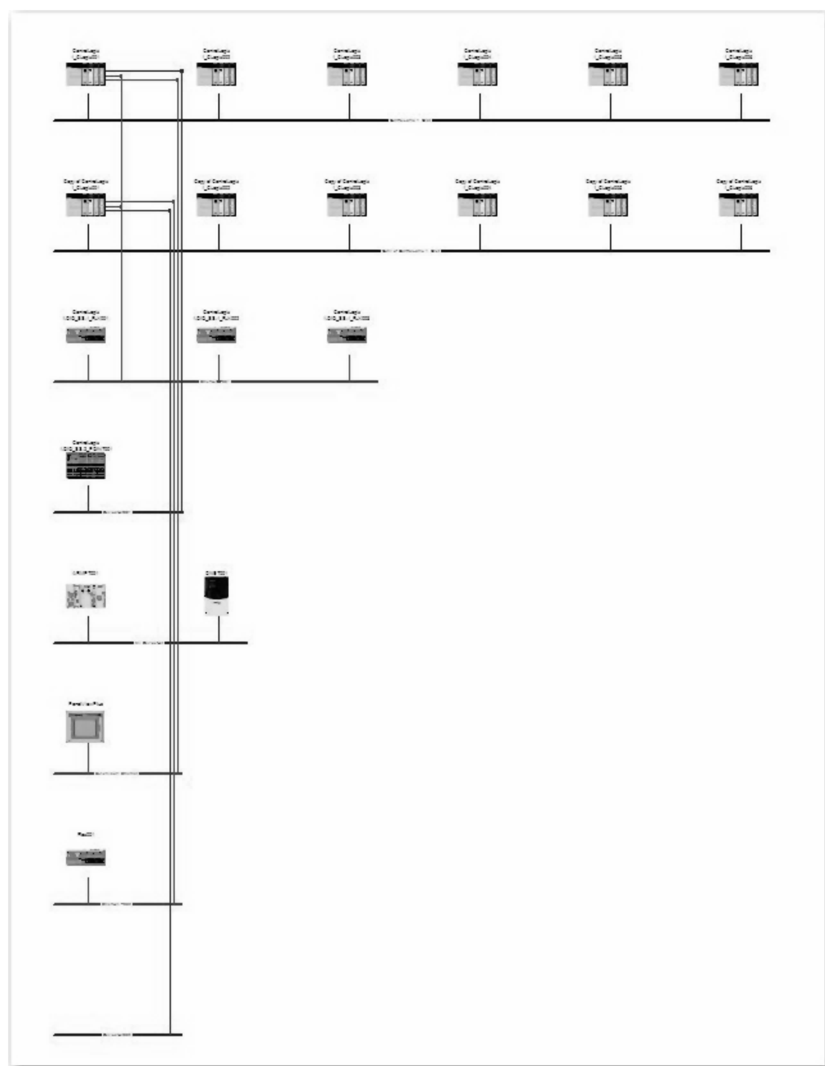


图 26-195　架构视图

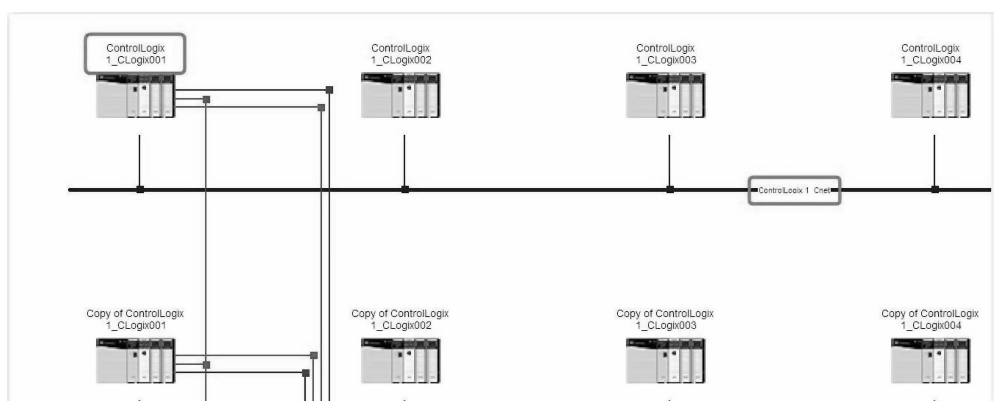


图 26-196　架构视图

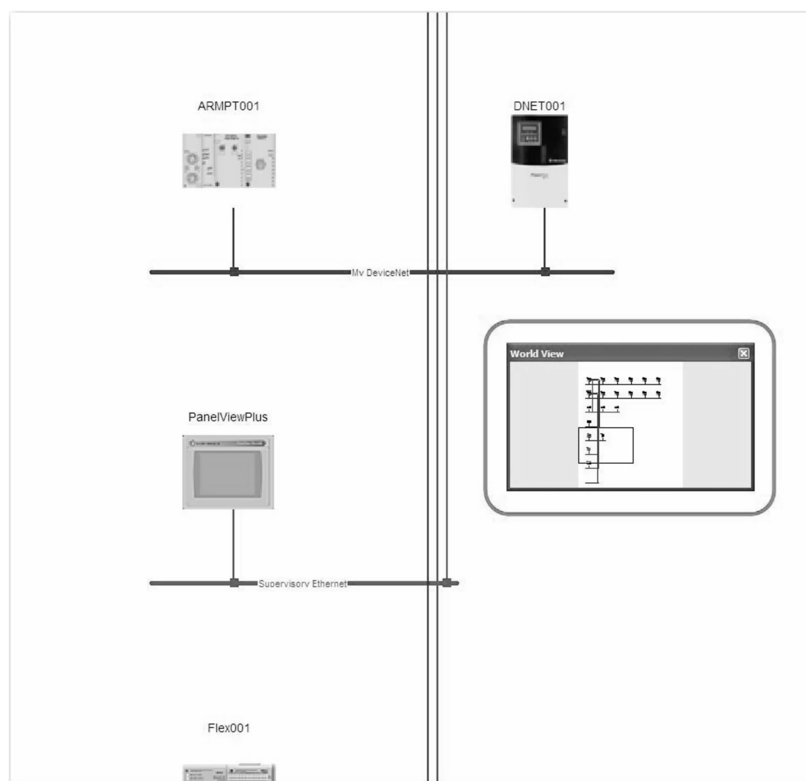


图 26-197　架构视图

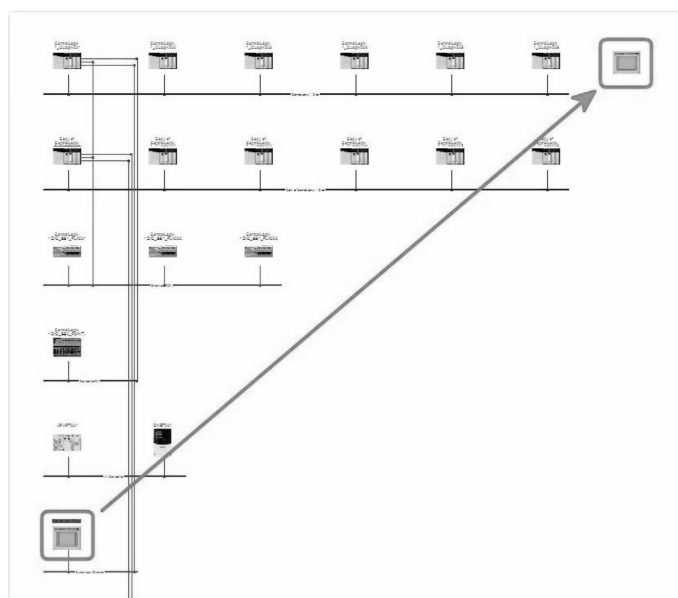


图 26-198　架构视图中移动设备

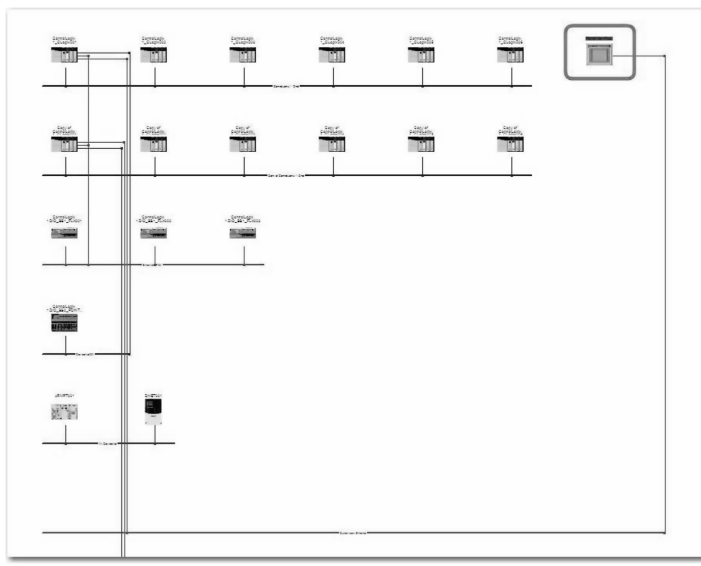


图 26-199 架构视图中移动设备

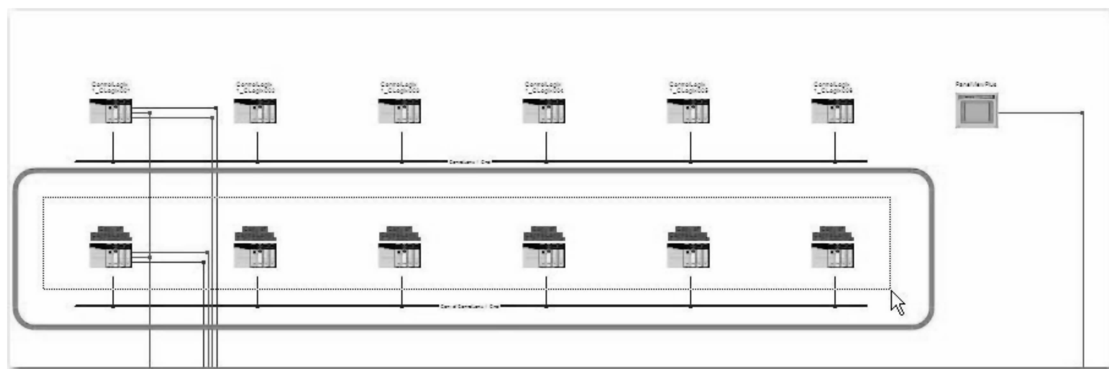


图 26-200 架构视图中拖动设备

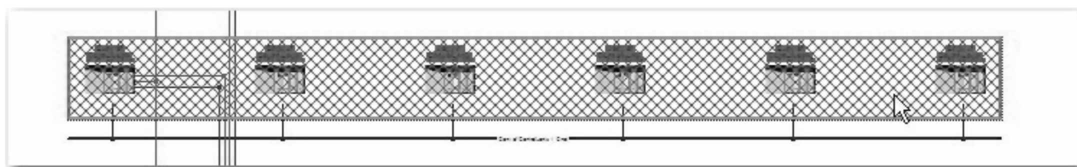


图 26-201 架构视图中拖动设备组



图 26-202　架构视图中拖动设备组

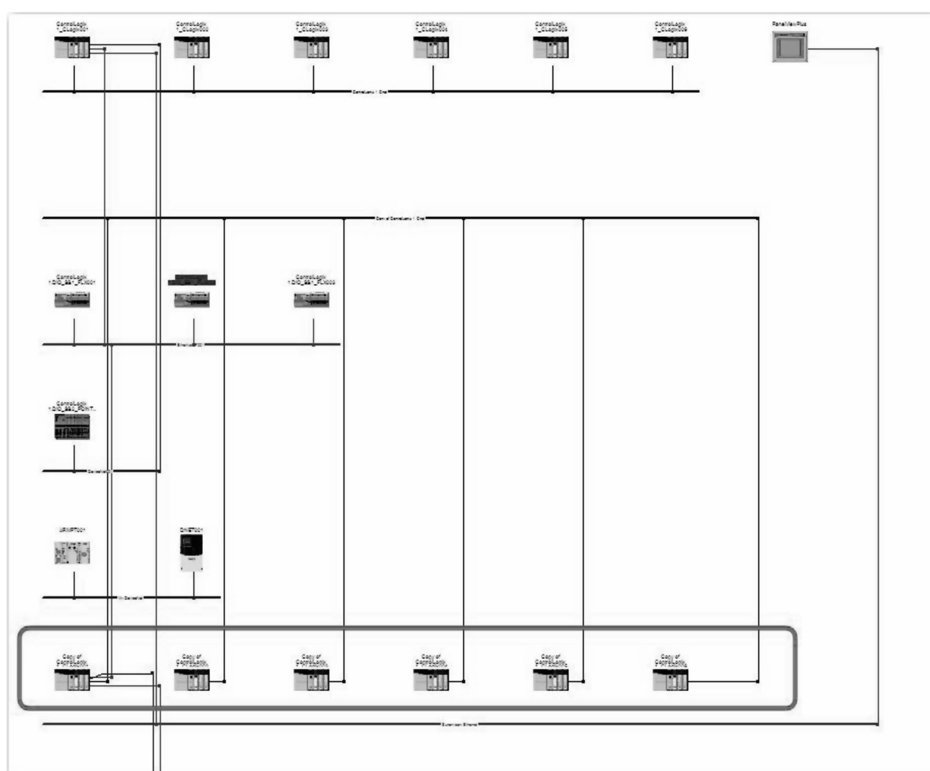


图 26-203　架构视图中拖动设备组

7. 也可“重新规划”或重画架构视图中的网络线路。在左侧的树列表中，右键单击 Supervisory Ethernet 项，然后选择“重新规划网络”（Repath Network），如图 26-204 所示。

视图现在显示的是间隔均匀的灰色框。可以在视图中的任何白色空间中重新规划网络，但不能在灰色空间中重新规划网络（灰色框是可以放置机架和设备的位置）。将光标（现在显示为十字形）置于视图上方线路下面的白色空间中，按住鼠标左键，将光标拖动到最后一个节点的右侧，然后释放鼠标按键，如图 26-205 所示。

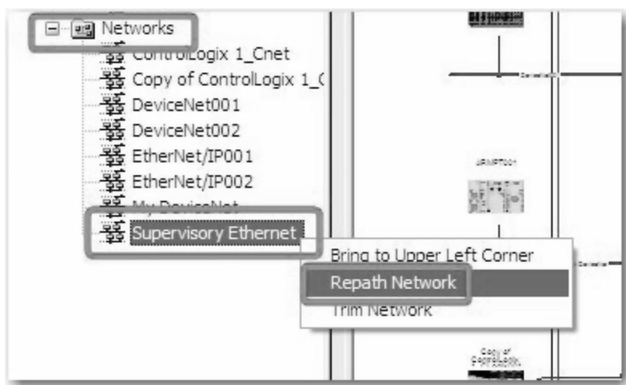


图 26-204 网络列表

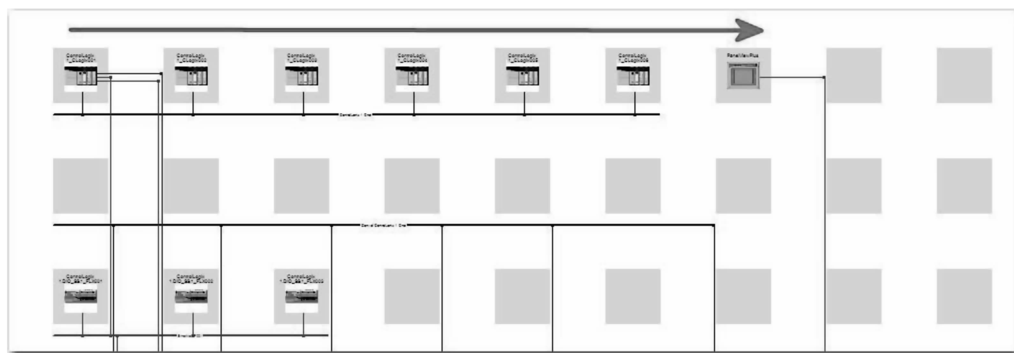


图 26-205 “重新规划网络”视图

IAB 会重画网络线路和连接，如图 26-206 所示。

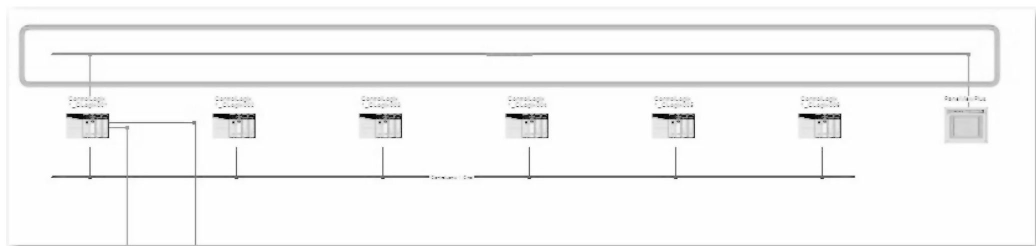


图 26-206“重新规划网络”视图

8. 可使用“封装和区域框”（Enclosure and Area box）光标将封装和区域框在架构视图中的项目附近拖动。可以使用“文本”（Text）按钮在架构视图中创建文本标签。在“架构视图”（Architecture View）工具栏中单击这些按钮，然后尝试拖动一些框并在视图中添加一些文本，如图 26-207 所示。

还可以尝试“调整网络”功能，以缩短延伸到其最后一个节点之外的网络的长度。

在架构视图的白色空间中右键单击。可以选择“打印架构视图”（Print Architecture View）。视图将打印在多张表单上。

十八、查看 BOM

IAB 中有两种 BOM：

1）快速 BOM：只显示活动窗口的 BOM。

2）项目 BOM：显示整个工作区的 BOM。在本练习中，将学会如何查看这两种 BOM。



图 26-207　“架构视图”工具栏（部分）

1. 单击“硬件”（Hardware）选项卡，在硬件列表中选择 ControlLogix 1_CLogix001，以显示该机架的硬件视图。单击“快速 BOM”（Quick BOM）按钮，如图 26-208 所示。

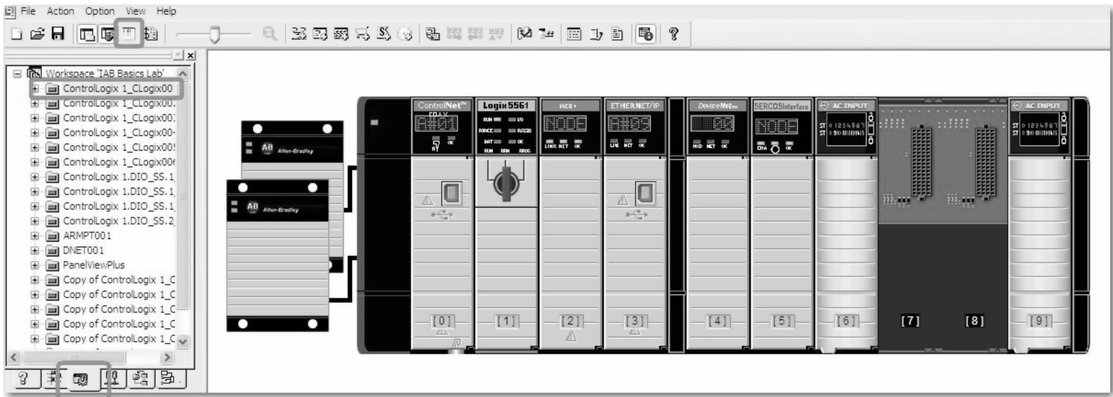


图 26-208　硬件列表

会出现该机架的“快速 BOM”（Quick BOM）。“快速 BOM”（Quick BOM）只会显示当前活动窗口中的项目，并且不会显示定价，如图 26-209 所示。

Bill of Material for Hardware : ControlLogix 1_CLogix001

Qty	Catalog #	Description
-08		Includes 1756-CPR2: Redundant power supply cables (Length = 0.9...
-08		Includes 1756-PSCA2: Redundant power supply chassis adapter mo...
001	1756-EN2T	EtherNet 10-100M Bridge Module
001	1756-EWEB	CLX Ethernet/IP Enhanced Web Server Module
001	1756-CN2	ControlNet Interface Module
001	1756-L61	Logix5561 Processor With 2Mbytes Memory
001	1756-M16SE	Servo Module, 16 Axis SERCOS
001	1756-PAR2	85-256 VAC Redundant Power Supply Assembly
001	1756-A10	1756 Chassis 10 slots
001	1756-DNB	DeviceNet Bridge/Scanner Module
002	1756-TBNH	20 Position NemA Screw Clamp Block
002	1756-IA16	79-132 VAC Input 16 Pts (20 Pin)

图 26-209　快速 BOM

单击“关闭”（Close），关闭“快速 BOM”（Quick BOM）窗口。

2. 关闭“快速 BOM”（Quick BOM）窗口并单击“项目 BOM”（Project BOM）按钮，如图 26-210 所示。

系统可能会询问：是否要保存对项目的更改。如果要保存，请单击“是”（Yes），如图 26-211 所示。

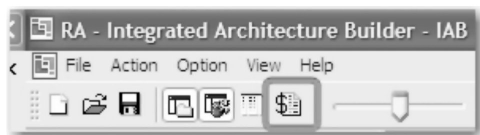


图 26-210 项目 BOM 按钮

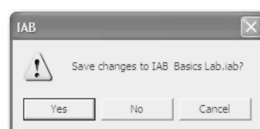


图 26-211 保存项目的更改

将出现该项目完整的“项目 BOM”（Project BOM）。项目中的所有部分以及定价信息将会列出（请完成价格文件实验练习，以了解 IAB 如何处理定价），如图 26-212 所示。

Project Bill Of Material

Qty	Catalog #	Description	Unit Price	Price
		System : New System		
		Subsystem of New System : ControlLogix 1		
006	1756-A10	1756 Chassis 10 slots	585.00	3510.00
006	1756-CN2	ControlNet Interface Module	2030.00	12180.00
001	1756-DNB	DeviceNet Bridge/Scanner Module	1120.00	1120.00
001	1756-EN2T	EtherNet 10-100M Bridge Module	2730.00	2730.00
001	1756-EWEB	CLX Ethernet/IP Enhanced Web Server Module	2620.00	2620.00
001	1756-HSC	2 Channel / 4 Output High Speed Counter (36 Pin)	1020.00	1020.00
009	1756-IA16	79-132 VAC Input 16 Pts (20 Pin)	441.00	3969.00
005	1756-IF16	Analog Input - Current/Voltage 16 Pts (36 Pin)	1480.00	7400.00
001	1756-L61	Logix5561 Processor With 2Mbytes Memory	5420.00	5420.00
001	1756-M16SE	Servo Module, 16 Axis SERCOS	3740.00	3740.00
009	1756-OA16	74-265 VAC Output 16 Pts (20 Pin)	622.00	5598.00
012	1756-OF6VI	Isolated Analog Output - Voltage 6 Pts (20 Pin)	2390.00	28680.00
005	1756-PA72	85-265 VAC Power Supply (5V @ 10 Amp)	867.00	4335.00
001	1756-PAR2	85-256 VAC Redundant Power Supply Assembly	2640.00	2640.00
		Includes 1756-PSCA2: Redundant power supply chassis adapter module. Qty - 1.		
		Includes 1756-CPR2: Redundant power supply cables (Length = 0.91m [3ft]). Qty - 2		
006	1756-TBCH	36 Pin Screw Clamp Block With Standard Housing	79.30	475.80
030	1756-TBNH	20 Position NemA Screw Clamp Block	63.90	1917.00
			Subtotal:	87354.80
		Network of ControlLogix 1 : ControlLogix 1_Cnet		
002	1786-XT	ControlNet BNC Terminator	9.27	18.54
006	1786-TPS	ControlNet Straight T-Tap	113.00	678.00
000	1786-RG6	RG-6 Quad Shield Coax Cable (5.0m reqd.)	628.00	
			Subtotal:	696.54
		Subsystem of New System : ControlLogix 1.DIO_SS.1		
003	1794-AENT	Ethernet Adapter Module	702.00	2106.00
002	1794-IA16	85-132V AC Input Module, 16 Point	381.00	762.00
002	1794-IB16	24V DC Sink Input Module, 16 Point	301.00	602.00
004	1794-IE8	Analog Input Module, 8 Point	952.00	3808.00
003	1794-OB8	24V DC Source Output Module, 8 Point	175.00	525.00
007	1794-OE4	Analog Output Module, 4 Point	862.00	6034.00
003	1794-OM8	159-264 VAC Output Module, 8 Point	331.00	993.00
003	1794-TB2	2-wire screw terminal base (16 I/O; 18 common; 2 +V)	105.00	315.00
015	1794-TB3	3-wire screw terminal base (16 I/O; 18 common; 18 +V)	190.00	2850.00
003	1794-TBNF	Fused NEMA terminal base (16 I/O; 2 common; 2 +V)	195.00	585.00
			Subtotal:	18580.00

System Status :
To be checked!

Save to XLS　Save to XML　Print　Close　Help

图 26-212 项目中的所有部分以及定价信息

请注意，可以通过 BOM 窗口底部的按钮打印 BOM 或将其导出为 Excel 或 XML 格式。单击 IAB 工具栏上的“保存”（Save）按钮，关闭 BOM 窗口并保存之前的操作。

十九、生成报表

1. 单击 IAB 工具栏中的“报表选项”（Report Options）按钮（或者在菜单栏中选择“选项 >

报表选项”（Option > Report Options）），如图 26-213 所示。



图 26-213　报表选项

会出现报表选项对话框，如图 26-214 所示。

在“报表选项”（Report Options）对话框中，可以设置以下内容：

1）模板文件（Template Files）：IAB 提供多个模板文件，这些模板文件会为其生成的报表数据提供结构。常规 IAB 项目（非过程）和过程项目都有单独的模板文件，并且必须指定这两种项目的模板文件。根据项目中主控制系统的不同，常规项目模板文件包含样板文件文本，并且会以多种语言表示。模板文件在 C：\ Program Files \ RA_IAB \ Input-Files 中提供。

单击浏览文件夹以查找模板文件。

单击红色 X 清空模板区域。

2）语言（Language）：指定标准报表部分以哪种语言表示。例如，每张报表都包含名为“总览”的部分，并且报表的每部分都包含一些标准文本项。这些是受“语言”选项影响的部分。该设备不会选择模板文件中使用的语言。必须分别选择模板文件和语言。

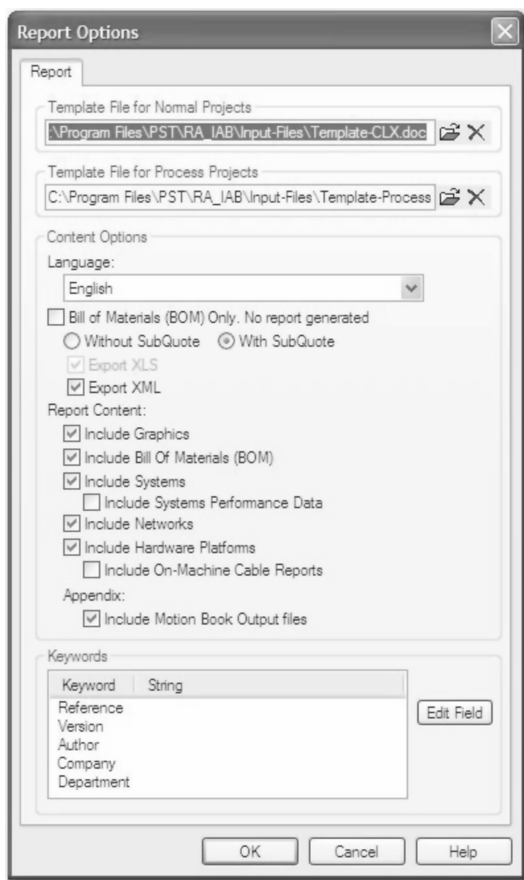


图 26-214　报表选项对话框

3）仅 BOM（BOM）：选中该框可仅输出物料清单。不会输出其他报表元素，其他元素的选择框会灰显。如果选中“导出 XLS”（Export XLS）框，则会以 Excel 格式输出 BOM；如果选中“导出 XML”（Export XML）框，则会以 XML 格式输出 BOM。带“子报价”表示将包含每个 BOM 部分的价格小计。不带子报价表示不会包含任何小计。即，将得到一张已合并的 BOM，且每个目录号占一行。

4）包含图像（Include Graphics）：报表将包含项目中的系统视图、网络视图和硬件视图。

5）包含物料清单（Include Bill of Materials）：报表将包含 BOM。如果选中该框，则“导出 XLS”（Export XLS）框会被选中并灰显，这样便无法对其进行更改。IAB 将生成 Excel 版本的 BOM，并将其包含在 Word 输出中。

6）包含系统（Include Systems）：报表将包含列出系统组件的 IAB 项目中每个系统的子部分。还将包含内存、连接和耗电量数据。

7）包含子系统性能数据（Include Subsystem Performance Data）：除了其他系统数据之外，报表还将包含每个子系统的向导推荐网络值。

8）包含网络（Include Networks）：报表将包含列出了项目中的所有网络及其节点的列表。

9）包含硬件平台（Include Hardware Platforms）：报表将包含列出了项目中所有硬件的列表。包括项目中的每个机架及其模块、每个 I/O 组件以及每台设备。

10）包含机旁电缆报表（Include On-Machine Cable Reports）：对于每个机旁布线解决方案，报表将包含列出电缆、连接器、分线盒等内容的列表。

11）包含运动控制书籍输出文件（Include Motion Book Output Files）：使用 Motion Analyzer 和 Motion Selector 进行运动控制组态时，运动控制技术参数文档会回到 IAB 中。如果选中该框，则会将 IAB 项目中的运动控制技术参数文档添加到报表中。

12）关键字（Keywords）：这些项目将出现在报表的封面页上。请参见下文以了解如何编辑这些字段。



图 26-215　输入作者名字

2. 在“关键字”（Keywords）区域中，选择“作者”（Author）字段，然后单击“编辑字段”（Edit Field）。在指示的位置键入名字，然后单击“确定”（OK），如图 26-215 所示。

3. 单击“报表”（Report）按钮。还可以选择“操作”（Action）菜单上的“创建报表”（Create Report），如图 26-216 所示。



图 26-216　报表按钮

4. 系统可能会询问：是否要保存对项目的更改。单击“是”（Yes），如图 26-217 所示。

5. 系统可能会告知：“常规检测器”中的警告。这是为了提醒，注意可能存在需要更正的内容。目前，请单击“是”（Yes），如图 26-218 所示。

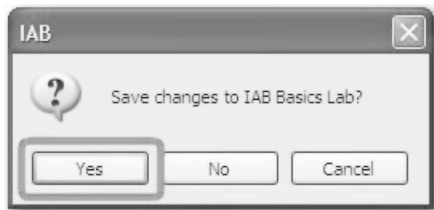


图 26-217　保存按钮

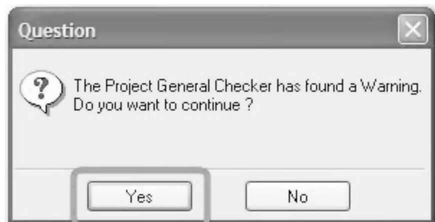


图 26-218　警告信息

6. IAB 将询问：要将报表保存在哪个位置。默认情况下，IAB 会选择包含 IAB 文件的项目文件夹。此时，可以指定任意目录。单击“保存”（Save）以使用项目文件夹，如图 26-219 所示。

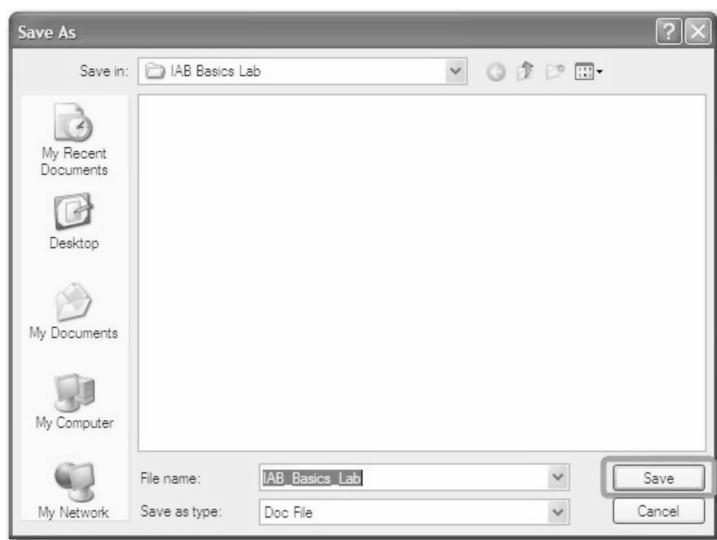


图 26-219　保存按钮

7. 可能会看到一条消息，提示 IAB 正在使用的价格文件中未找到某些项目。单击“确定”（OK）接受该消息，如图 26-220 所示。



图 26-220　接受该消息

8. 将出现声明。单击“确定”（OK），如图 26-221 所示。

9. IAB 会处理报表，并询问：要打开哪种报表。选择“使用 Word 应用程序的完整报表?”（The full report using your Word application?），然后单击“是”（Yes），如图 26-222 所示。

会在 Microsoft Word 中打开报表。检查封面页，并查看的名字是否显示为作者。检查报告的其他内容。

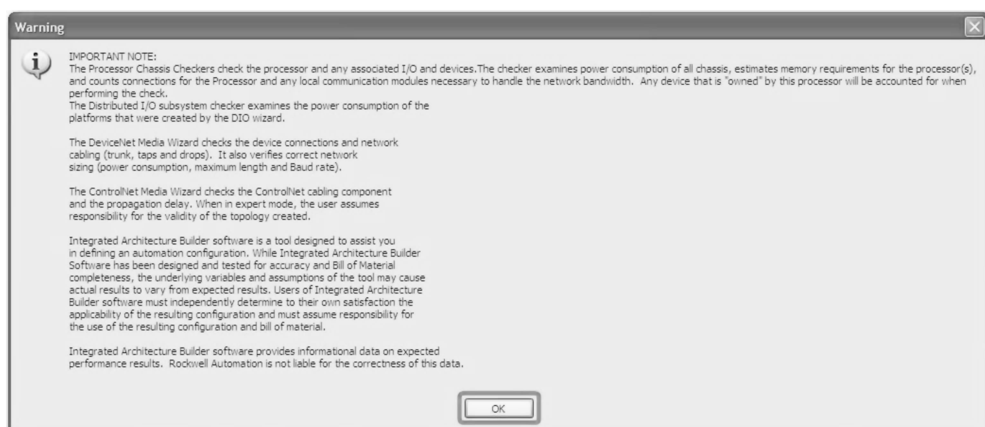
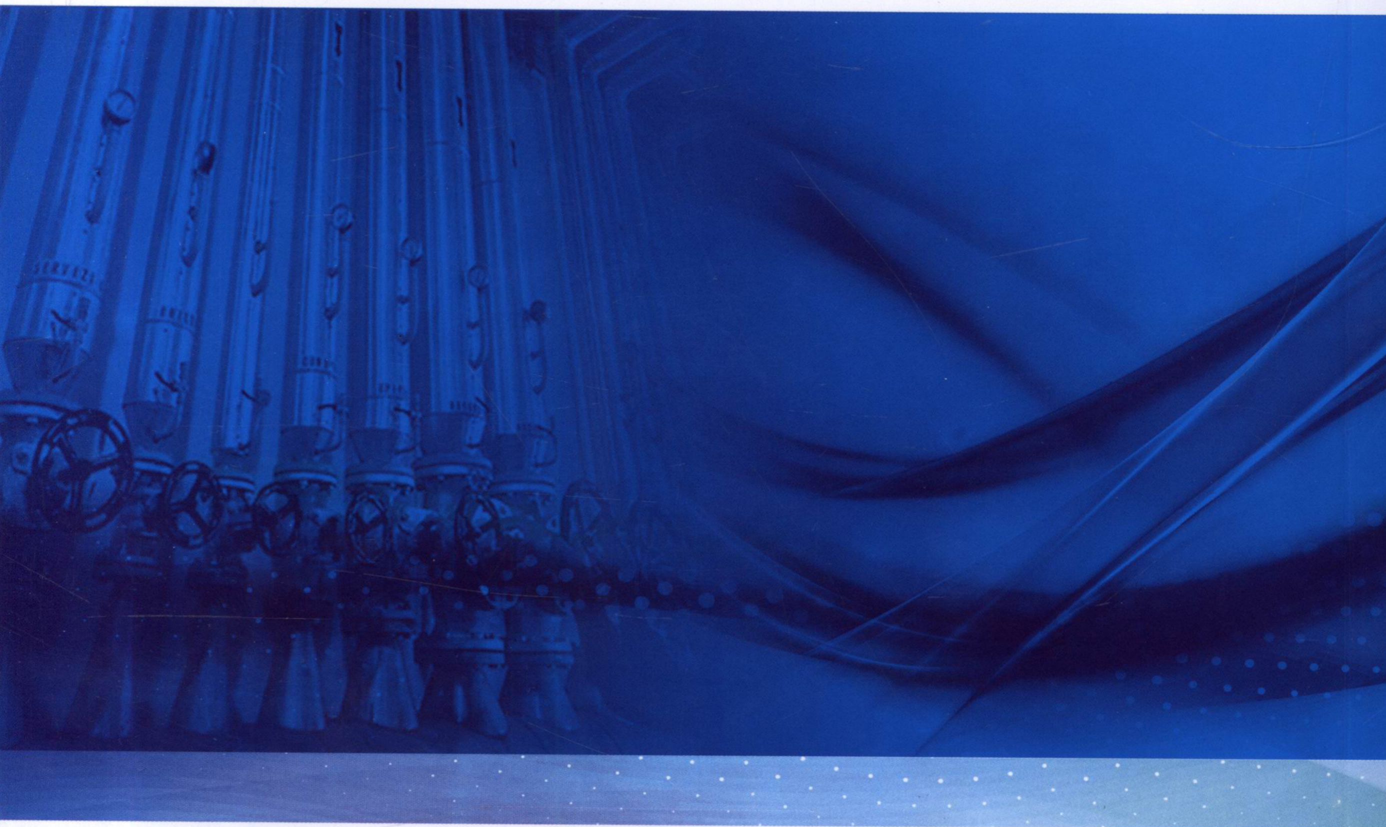


图 26-221　确定声明信息



图 26-222　选择处理报表



地址：北京市百万庄大街22号
邮政编码：100037
电话服务
社服务中心：010-88361066
销售一部：010-68326294
销售二部：010-88379649
读者购书热线：010-88379203
网络服务
教材网：http://www.cmpedu.com
机工官网：http://www.cmpbook.com
机工官博：http://weibo.com/cmp1952
封面无防伪标均为盗版

ISBN 978-7-111-47798-3
策划编辑◎林春泉



定价：59.00元